2008-01-01

Multi-algorithmic Cryptography using Deterministic Chaos with Applications to Mobile Communications

Jonathan Blackledge
Technological University Dublin, jonathan.blackledge@tudublin.ie

Recommended Citation

Blackledge, J.: Multi-algorithmic Cryptography using Deterministic Chaos with Applications to Mobile Communications. ISAST Transactions on Electronics and Signal Processing. ISSN 1797-2329, issue: No. 1, Vol. 2, pages: 23-64, 2008. doi:10.21427/D7VH04

Multi-algorithmic Cryptography using Deterministic Chaos with Applications to Mobile Communications

Jonathan M Blackledge, Fellow, IET, Fellow, BCS, Fellow, IMA, Fellow, RSS

***Abstract*— In this extended paper, we present an overview of the principal issues associated with cryptography, providing historically significant examples for illustrative purposes as part of a short tutorial for readers that are not familiar with the subject matter. This is used to introduce the role that nonlinear dynamics and chaos play in the design of encryption engines which utilize different types of Iteration Function Systems (IFS). The design of such encryption engines requires that they conform to the principles associated with diffusion and confusion for generating ciphers that are of a maximum entropy type. For this reason, the role of confusion and diffusion in cryptography is discussed giving a design guide to the construction of ciphers that are based on the use of IFS. We then present the background and operating framework associated with a new product - Crypstic^{TM} - which is based on the application of multi-algorithmic IFS to design encryption engines mounted on a USB memory stick using both disinformation and obfuscation to 'hide' a forensically inert application. The protocols and procedures associated with the use of this product are also briefly discussed.**

***Index Terms*— Cryptography, Nonlinear Dynamics, Iteration Function Systems, Chaos, Multi-algorithmicity**

I. Introduction

THE quest for inventing innovative techniques which only allow authorized users to transfer information that is impervious to attack by others has, and continues to be, an essential requirement in the communications industry (e.g. [1], [2], [3]). This requirement is based on the importance of keeping certain information secure, obvious examples being military communications and financial transactions, the former example, being a common theme in the history and development of Cryptology [4], [5], [6].

Cryptography is the study of mathematical and computational techniques related to aspects of information security (e.g. [7]-[9]). The word is derived from the Greek *Kryptos*, meaning hidden, and is related to disciplines such as *Cryptanalysis* and *Cryptology*. Cryptanalysis is the art of breaking cryptosystems by developing techniques for the retrieval of information from encrypted data without having *a priori* knowledge of the required decryption process (typically, knowledge of the key) [10]. Cryptology is the science that underpins Cryptography and Cryptanalysis and can include a broad range of mathematical concepts, computational algorithms and technologies. In other words, Cryptology is a multi-disciplinary subject that covers a wide spectrum of different disciplines and increasingly involves using a range of engineering concepts and technologies through the innovation associated with term 'technology transfer'. These include areas such as Synergetics, which is an interdisciplinary science explaining the formation and self-organization of patterns and structures in non-equilibrium open systems and Semiotics, which is the study of both individual and grouped signs and symbols, including the study of how meaning is constructed and understood [11].

Cryptology is often concerned with the application of formal mathematical techniques to design a cryptosystem and to estimate its theoretical security. This can include the use of formal methods for the design of security software which should ideally be a 'safety critical' [12]. However, although the mathematically defined and provable strength of a cryptographic algorithm or cryptosystem is necessary, it is not a sufficient requirement for a system to be acceptably secure. This is because it is difficult to estimate the security of a cryptosystem in any formal sense when it is implemented under operational conditions that cannot always be predicted and thus, simulated. The security associated with a cryptosystem can be checked only by means of proving its resistance to various kinds of known attack that are likely to be implemented. However, in practice, this does not mean that the system is secure since other attacks may exist that are not included in simulated or test conditions. The reason for this is that humans possess a broad range of abilities from unbelievable ineptitude to astonishing brilliance which can not be formalized in a mathematical sense or on a case by case basis.

The practical realities associated with Cryptology are indicative of the fact that 'security is a process, not a product' [13]. Whatever the sophistication of the security product (e.g. the encryption and/or key exchange algorithm(s), for example), unless the user adheres strictly to the procedures and protocols designed for its use the 'product' can be severely compromised. A good example of this is the use of the Enigma [14] cipher by Germany during the Second World War. It was not just the 'intelligence' of the 'code breakers' at Bletchley Park in England that allowed the allies to break many of the Enigma codes but the 'irresponsibility' and, in many cases, the sheer stupidity of the way in which the system was used by the German armed and intelligence services at the time.

The basic mechanism for the Enigma cipher, which had been developed as early as 1923 by Artur Schubius for se-

Manuscript received November 1, 2007. The work reported in this paper has been supported by the Council for Science and Technology, Management and Personnel Services Limited and Crypstic^{TM} Limited

Jonathan Blackledge (jon.blackledge@btconnect.com) is Visiting Professor, Department of Electronic and Electrical Engineering, Loughborough University, England (http://www.lboro.ac.uk/departments/el/staff/blackledge.html) and Extraordinary Professor, Department of Computer Science, University of the Western Cape, Cape Town, Republic of South Africa (http://www.cs.uwc.ac.za).

curing financial transactions, was well known to the allies due primarily to the efforts of the Polish Cipher Office at Poznan in the 1930s. The distribution of some 10,000 similar machines to the German army, navy and air force was therefore a 'problem waiting to happen'. The solution would have been to design a brand new encryption engine or better still, a range of different encryption engines given the technology of the time, and use the Enigma machine to propagate disinformation. Indeed, some of the new encryption engines introduced by the Germans towards the end of the Second World War were not broken by the allies.

These historically intriguing insights are easy to contemplate in hindsight, but they can also help to focus on the methodologies associated with developing new technologies for knowledge management which is a focus of the material considered in this work. Here, we explore the use of deterministic chaos for designing ciphers that are composed of many different pseudo chaotic number generating algorithms - Meta-encryption-engines. This multi-algorithmic or Meta-engine approach provides a way of designing an unlimited class of encryption engines as opposed to designing a single encryption engine that is operated by changing the key(s) - which for some systems, public key system in particular, involves the use of prime numbers. There are of course a number of disadvantages to this approach which are discussed later on but it is worth stating at this point, that the principal purpose for exploring the application of deterministic chaos in cryptography is:

- the non-reliance of such systems on the use of prime numbers which place certain limits on the characteristics and arithmetic associated with an encryption algorithm;
- the potentially unlimited number of chaos based algorithms that can be, quite literally, invented to produce a meta-encryption engine.

II. Information and Knowledge Management

With regard to information security and the management of information in general, there are some basic concepts that are easy to grasp and sometimes tend to get lost in the detail. The first of these is that the recipient of any encrypted message must have some form of *a priori* knowledge on the method (the algorithm, for example) and the operational conditions (e.g. the key) used to encrypt a message. Otherwise, the recipient is in no better 'state of preparation' than the potential attacker. The conventional approach is to keep this *a priori* information to a minimum but in such a way that it is critical to the decryption process. Another important reality is that in an attack, if the information transmitted is not deciphered in good time, then it may become redundant. Coupled with the fact that an attack usually has to focus on a particular criterion (such as a specific algorithm), one way to enhance the security of a communications channel is to continually change the encryption algorithm and/or process offered by the technology available.

Another approach to information management is to disguise or camouflage the encrypted message in what would appear to be 'innocent' or 'insignificant' data such as a digital photograph, a music file or both, for example[1]. This is known as *Steganography* [15]-[17]. Further, the information security products themselves should be introduced and 'organised' in such a way as to reflect their apparent insignificance in terms of both public awareness and financial reward which helps to combat the growing ability to 'hack and crack' using increasingly sophisticated software that is readily available, e.g. [18]. This is of course contrary to the dissemination of many encryption systems, a process that is commonly perceived as being necessary for business development through the establishment of a commercial organisation, international patents, distribution of marketing material, elaborate and sophisticated Web sites, authoritative statements on the strength of a system to impress customers, publications and so on. Thus, a relatively simple but often effective way of maintaining security with regard to the use of an encryption system is to not tell anyone about it. The effect of this can be enhanced by publishing other systems and products that are designed to mislead the potential attacker. In this sense, information management and Information and Communication Technology (ICT) security products should be treated in the same way as many organisations treat a breach of security, i.e. not to publish the breach in order to avoid embarrassment and loss of faith by the client base.

A. Secrets and Ultra-secrets

A classic mistake (of historical importance) of not 'keeping it quiet', in particular, not maintaining 'silent warfare' [19], was made by Winston Churchill when he published his analysis of World War I. In his book, *The World Crisis 1911-1918*, published in 1923, he stated that the British had deciphered the German Naval codes for much of the war as a result of the Russians salvaging a code book from the small cruiser *Magdeburg* that had ran aground off Estonia on August 27, 1914. The code book was passed on to Churchill who was, at the time, the First Sea Lord. This helped the British maintain their defences with regard to the German navy before and after the Battle of Jutland in May, 1916. The German navy became impotent which forced Germany into a policy of unrestricted submarine warfare. In turn, this led to an event (the sinking on May 7, 1915 of the Lusitania, torpedoed by a German submarine, the U-20) that galvanized American opinion against Germany and played a key role in the United States' later entry into the Fisrt World War on April 17, 1917 and the defeat of Germany [20], [21].

Churchill's publication did not go un-noticed by the German military between the First and Second World Wars. Consequently, significant efforts were made to develop new encryption devices for military communications. This resulted in the famous Enigma machine, named after Sir Edward Elgar's masterpiece, the *Enigma Variations* [22]. Enigma was an electro-mechanical machine about the size of a portable typewriter which, through application of both electrical ('plug-board') and mechanical ('rotor') settings offered $\sim 2 \times 10^{20}$ permutations for establishing a 'key'. The machine could be

[1]By encoding the encrypted message in the least significant bit or bit-pair of the host data, for example.

used without difficulty by semi-skilled operators under the most extreme battle conditions. The keys could be changed daily or several times a day according to the number of messages transmitted.

The interest in cryptology by Germany that was undoubtedly stimulated by Churchill's indiscretions included establishing a specialist cipher school in Berlin. Ironically, it was at this School that some of the Polish mathematicians were trained who later worked for the Polish Cipher Office, opened in utmost secrecy at Poznan in 1930 [23], [24]. In January 1929, the Dean of the Department of Mathematics, Professor Zdzislaw Krygowski from the University of Poznan, provided a list of his best graduates to start working at this office. One of these graduates was the brilliant young logician, Marian Rejewski who pioneered the design of the *Bomba kryptologiczna*, an electro-mechanical device used for eliminating combinations that had not been used to encrypt a message with the Enigma cipher [25]. However, the design of the *Bomba kryptologiczna* was only made possible through the Poles gaining access to the Enigma machine and obtaining knowledge of its mechanism without alerting the Germans to their activities. In modern terms, this is equivalent to obtaining information on the type of encryption algorithm used in a cryptosystem.

The *Bomba kryptologiczna* helped the Poles to decipher some 100,000 Enigma messages from as early January 1933 to September 1939 including details associated with the remilitarization of the Rhine Province, Anschluss of Austria and seizure of the Sudetenland. It was Rejewski's original work that formed the basis for designing the advanced electro-mechanical and later, the electronic decipher machines (including 'Colossus' - the world's first programmable computer) constructed and utilized at Bletchley Park between 1943 and 1945 [26], [27].

After the Second World War, Winston Churchill made sure that he did not repeat his mistake, and what he referred to as his 'Ultra-secret' - the code breaking activities undertaken at Station X in Bletchley Park, England - was ordered by him to be closed down and the technology destroyed soon after the end of the war. Further, Churchill never referred to his Ultra-secret in any of his publications after the war. Those personnel who worked at Bletchley Park were required to maintain their silence for some fifty years afterwards and some of the activities at Bletchley Park remain classified to this day. Bletchley Park is now a museum which includes a reconstruction of 'Colossus' undertaken in the mid-1990s. However, the type of work undertaken there in the early 1940s continues in many organisations throughout the world such as the Government Communications Head Quarters (GCHQ) based at Cheltenham in England [28] where a range of 'code making' and 'code breaking' activities continue to be developed.

The historical example given above clearly illustrates the importance of maintaining a level of secrecy when undertaking cryptographic activities. It also demonstrates the importance of not publishing new algorithms, a principle that is at odds with the academic community; namely, that the security of a cryptosystem should not depend upon algorithm secrecy. However, this has to be balanced with regard to the dissemination of information in order to advance a concept through peer review, national and international collaboration. Taken to an extreme, the secrecy factor can produce a psychological imbalance that is detrimental to progress. Some individuals like to use confidential information to enhance their status. In business, this often leads to issues over the signing of Non-Disclosure Agreements or NDAs, for example, leading to delays that are of little value, especially when it turns out that there is nothing worth disclosing. Thus, the whole issue of 'keeping it quiet' has to be implemented in a way that is balanced, such that confidentiality does not lead to stagnation in the technical development of a cryptosystem. However, used correctly and through the appropriate personality, issues over confidentiality coupled with the 'feel important' factor can be used to good effect in the dissemination of disinformation.

B. Home-Spun Systems Development

The development and public use of information security technology is one of the most interesting challenges for state control over the 'information society'. As more and more members of the younger generation become increasingly IT literate, it is inevitable that a larger body of perfectly able minds will become aware of the fact that cryptology is not as difficult as they may have been led to believe. As with information itself, the days when cryptology was in the hands of a select few with impressive academic credentials and/or luxury civil service careers are over and cryptosystems can now be developed by those with a diverse portfolio of backgrounds which does not necessarily include a University education. This is reflected in the fact that after the Cold War, the UK Ministry of Defence, for example, developed a strategy for developing products driven by commercially available systems. This Commercial-Off-The-Shelf or COTS approach to defence technology has led directly to the downsizing of the UK Scientific Civil Service which, during the Cold War, was a major source of scientific and technical innovation.

The average graduate of today can rapidly develop the ability to write an encryption system which, although relatively simple, possibly trivial and ill-informed, can, by the very nature of its non-compliance to international standards, provide surprisingly good security. This can lead to problems with the control and management of information when increasingly more individuals, groups, companies, agencies and nation states decide that they can 'go it alone' and do it themselves. While each home grown encryption system may be relatively weak, compared to those that have had expert development over many years, have been well financed and been tested against the very best of attack strategies, the proliferation of such systems is itself a source of significant difficulty for any authority whose role is to monitor communications traffic in a way that is timely and cost effective. This is why governments world-wide are constantly attempting to control the use and exploitation of new encryption methods in the commercial sector[2]. It also explains the importance of international encryption standards in terms of both public perception and

[2]For example, the introduction of legislation in mainland UK concerning the decryption of messages by a company client through enforcement of the Regulation of Investigatory Powers (RIP) Act, 2000.

free market exploitation. Government and other controlling authorities like to preside over a situation in which everybody else is confidently reliant for their information security on products that have been developed by the very authorities that encourage their use, a use that is covertly 'diffused' into the 'information society' through various legitimate business ventures coupled with all the usual commercial sophistication and investment portfolios. Analysis of this type can lead to a range of unsubstantiated conspiracy theories, but it is only by thinking through such possible scenarios, that new concepts in information management, some of which may be of practical value, are evolved. The proliferation of stand-alone encryption systems that are designed and used by informed individuals is not only possible but inevitable, an inevitability that is guided by the principle that if you want to know what you are eating then you should cook it yourself. Security issues of this type have become the single most important agenda for future government policy on information technology, especially when such systems have been 'home spun' by those who have learned to fully respect that they should, in the words of Shakespeare, 'Neither a borrower, nor a lender be'[3].

C. Disinformation

Disinformation is used to tempt the 'enemy' into believing certain kinds of information. The information may not be true or contain aspects that are designed to cause the enemy to react in an identifiable way that provides a strategic advantage [29], [30]. Camouflage, for example, is a simple example of disinformation [31]. This includes techniques for transforming encrypted data into forms that resemble the environments through which an encrypted message is to be sent [32], [33]. At a more sophisticated level, disinformation can include encrypted messages that are created with the sole purpose of being broken in order to reveal information that the enemy will react to by design.

Disinformation includes arranging events and processes that are composed to protect against an enemy acquiring knowledge of a successful encryption technology and/or a successful attack strategy. A historically significant example of this involved the Battle of Crete which began on the morning of 20 May 1941 when Nazi Germany launched an airborne invasion of Crete under the code-name Unternehmen Merkur (Operation Mercury) [34]. During the next day, through miscommunication and the failure of Allied commanders to grasp the situation, the Maleme airfield in western Crete fell to the Germans which enabled them to fly in heavy reinforcements and overwhelm the Allied forces. This battle was unique in two respects: it was the first airborne invasion in history[4]; it was the first time the Allies made significant use of their ability to read Enigma codes. The British had known for some weeks prior to the invasion of Crete that an invasion was likely because of the work being undertaken at Bletchley Park. They faced a problem because of this. If Crete was reinforced in order to repel the invasion then Germany would suspect that their encrypted communications were being compromised. But this would also be the case if the British and other Allied troops stationed on Crete were evacuated. The decision was, therefore, taken by Churchill to let the German invasion proceed with success but not without giving the invaders a 'bloody nose'. Indeed, in light of the heavy casualties suffered by the parachutists, Hitler forbade further airborne operations and Crete was dubbed 'the graveyard of the German parachutists'. The graveyard for German, British, Greek and Allied soldiers alike was not a product of a fight over desirable and strategically important territory (at least for the British). It was a product of the need to secure Churchill's 'Ultra-secret'. In other words, the Allied efforts to repulse the German invasion of Crete was, in reality, a form of disinformation, designed to secure a secret that was, in the bigger picture, more important than the estimated 16,800 dead and wounded that the battle cost.

[3]From William Shakespeare's play, Hamlet.

[4]Illustrating the potential of paratroopers and so initiating the Allied development of their own airborne divisions.

D. Plausible Deniability

Deniable encryption allows an encrypted message to be decrypted in such a way that different and plausible plaintexts can be obtained using different keys [35]. The idea is to make it impossible for an attacker to prove the existence of the real message, a message that requires a specific key. This approach provides the user with a solution to the 'gun to the head problem' as it allows the sender to have plausible deniability if compelled to give up the encryption key.

There are a range of different methods that can be designed to implement such a scheme. For example, a single ciphertext can be generated that is composed of randomised segments or blocks of data which correlate to blocks of different plaintexts encrypted using different keys. A further key is then required to assemble the appropriate blocks in order to generate the desired decrypt. This approach, however, leads to ciphertext files that are significantly larger than the plaintexts they contain. On the other hand, a ciphertext file should not necessarily be the same size as the plaintext file and padding out the plaintext before encryption can be used to increase the Entropy of the ciphertext (as discussed in Section VIII).

Other methods used for deniable encryption involve establishing a number of abstract 'layers' that are decrypted to yield different plaintexts for different keys. Some of these layers are designed to include so-called 'chaff layers'. These are layers that are composed of random data which provide the owner of the data to have plausible deniability of the existence of layers containing the real ciphertext data. The user can store 'decoy files' on one or more layers while denying the existence of others, identifying the existence of chaff layers as required. The layers are based on file systems that are typically stored in a single directory consisting of files with filenames that are either randomized (in the case where they belong to chaff layers), or are based on strings that identify cryptographic data, the timestamps of all files being randomized throughout.

E. Obfuscation

In a standard computing (windows) environment, a simple form of camouflage can be implemented by renaming files to

be of a different type; for example, storing an encrypted data file as a *.exe* or *.dll* file. Some cryptosystems output files with identifiable extensions such as .enc which can then be simply filtered by a firewall. Another example includes renaming files in order to access data and/or execute an encryption engine. For example, by storing an executable file as a *.dll* (dynamic link library) file (which has a similar structure to a *.exe* file) in a directory full of real *.dll* files associated with some complex applications package, the encryption engine can be obfuscated, especially if it has a name that is similar to the environment of files in which it is placed. By renaming the file back to its 'former self', execution of a cryptosystem can be undertaken in the usual way. However, this requires that the executable file is forensically inert, i.e. it does not contain data that reflects its purpose. A simple way of implementing this requirement is to ensure that the source code (prior to compilation) is devoid of any arrays, comments etc. that include references (through use of named variables, for example) to the type of application (e.g. comments such as *encrypt the data* or named arrays such as *decrypt_array*[i]).

F. Steganographic Encryption

It is arguable that disinformation should, where possible, be used in conjunction with the exchange of encrypted information which has been camouflaged using steganographic techniques for hiding the ciphertext. For example, suppose that it had been known by Germany that the Enigma ciphers were being compromised by the British during the Second World War. Clearly, it would have then been strategically advantageous for Germany to propagate disinformation using Enigma. If, in addition, 'real information' had been encrypted differently and the ciphertexts camouflaged using broadcasts through the German home radio service, for example, then the outcome of the war could have been very different. The use of new encryption methods coupled with camouflage and disinformation, all of which are dynamic processes, provides a model that, while not always of practical value, is strategically comprehensive and has only rarely been fully realised. Nevertheless, some of the techniques that have been developed and are reported in this work are the result of an attempt to realise this model.

III. Basic Concepts

Irrespective of the wealth of computational techniques that can be invented to encrypt data, there are some basic concepts that are a common theme in modern cryptography. The application of these concepts typically involves the use of random number generators and/or the use of algorithms that originally evolved for the generation of random number streams, algorithms that are dominated by two fundamental and interrelated themes [4]-[6]: (i) the use of modular arithmetic; (ii) the application of prime numbers. The application of prime numbers is absolutely fundamental to a large range of encryption processes and international standards such as PKI (Public Key Infrastructure) details of which are discussed later.

Fig. 1. Alice and Bob can place a message in a box which can be secured using a combination lock and sent via a public network - the postal service, for example.

Using a traditional paradigm, we consider the problem of how Alice (A) and Bob (B) can pass a message to and from each other without it being compromised or 'attacked' by an intercept. As illustrated in Figure 1, we consider a simple box and combination lock scenario. Alice and Bob can write a message, place it in the box, lock the box and then send it through an open 'channel' - the postal services, for example. In cryptography, the strength of the box is analogous to the strength of the cipher. If the box is 'weak' enough to be opened by brute force, then the strength of the lock is relatively insignificant. This is analogous to a cipher whose statistical properties are poor, for example, i.e. whose Probability Density Function (PDF) is narrow and whose information Entropy is relatively low, with a similar value to the plaintext. The strength of the lock is analogous to the strength of the key in a real cryptographic system. This includes the size of the combination number which is equivalent to the length of the key that is used. Clearly a four rotor combination lock as illustrated in Figure 1 represents a very weak key since the number of ordered combinations required to attempt a brute force attack to open the lock are relatively low, i.e. for a 4-digit combination lock where each rotor has ten digits 0-9, the number of possible combinations is 10000 (including 0000). However, the box-and-lock paradigm being used here is for illustrative purposes only.

A. Symmetric Encryption

Symmetric encryption is the simplest and most obvious approach for Alice and Bob to send messages. Alice and Bob agree on a combination number *a priori*. Alice writes a message, puts it in the box, locks it and sends it off. Upon receipt, Bob unlocks the box using the combination number that has been agreed and recovers the message. Similarly, Bob can send a message to Alice using exactly the same approach or 'protocol'. Since this protocol is exactly the same for Alice and Bob it has a symmetry and thus, encryption methods that adopt this protocol are referred to as symmetric encryption methods. Given that the box and the lock have been designed to be strong, the principal weakness associated with this method is its vulnerability to attack if a third party obtains the combination number at the point when Alice and

Bob invent it and agree upon it. Thus, the principal problem in symmetric encryption is how Alice and Bob exchange the key. Irrespective of how strong the cipher and key are, unless the key exchange problem can be solved in an appropriate and a practicable way, symmetric encryption always suffers from the same fundamental problem - key exchange!

If E denotes the encryption algorithm that is used which depends upon a key K to encrypt plaintext P, then we can consider the ciphertext C to be given by

$$C = E_K(P).$$

Decryption can then be denoted by the equation

$$P = E_K(C).$$

Note that it is possible to encrypt a number of times using different keys $K_1, K_2, ...$ with the same encryption algorithm to give a double encrypted cipher text

$$C = E_{K_2}(E_{K_1}(P))$$

or a triple encrypted ciphertext

$$C = E_{K_3}(E_{K_2}(E_{K_1}(P))).$$

Decryption, is then undertaken using the same keys in the reverse order to which they have been applied, i.e.

$$P = E_{K_1}(E_{K_2}(E_{K_3}(C))).$$

Symmetric encryption systems, which are also referred to as shared secret systems or private key systems, are usually significantly easier to use than systems that employ different protocols (such as asymmetric encryption). However, the requirements and methods associated with key exchange sometimes make symmetric systems difficult to use. Examples of symmetric encryption systems include the Digital Encryption Standard DES and DES3 (essentially, but not literally, the Digital Encryption Standard with triple encryption) and the Advanced Encryption Standard (AES). Symmetric systems are commonly used in many banking and other financial institutes and in some military applications. A well known historical example of a symmetric encryption engine, originally designed for securing financial transactions, and later used for military communications, was the *Enigma*.

B. Asymmetric Ciphers

Instead of Alice and Bob agreeing on a combination number *a priori*, suppose that Alice sets her lock to be open with a combination number known only to her. If Bob then wishes to send Alice a message, he can make a request for her to send him an open lock. Bob can then write his message, place it in the box which is then locked and sent on to Alice. Alice can then unlock the box and recover the message using the combination number known only to her. The point here is that Bob does not need to know the combination number, he only needs to receive an open lock from Alice. Of course Bob can undertake exactly the same procedure in order to receive a message from Alice. Clearly, the processes that are undertaken by Alice and Bob in order to send and receive a single message are not the same. The protocol is asymmetric and we refer to encryption systems that use this protocol as being asymmetric. Note that Alice could use this protocol to receive messages from any number of senders provided they can get access to one of her open locks. This can be achieved by Alice distributing many such locks as required.

One of the principal weaknesses of this approach relates to the lock being obtained by a third party whose interest is in sending bogus or disinformation to Alice. The problem for Alice is to find a way of validating that a message sent from Bob (or anyone else who is entitled to send messages to her) is genuine, i.e. that the message is authentic. Thus, data authentication becomes of particular importance when implementing asymmetric encryption systems.

Asymmetric encryption relies on both parties having two keys. The first key (the public key) is shared publicly. The second key is private, and is kept secret. When working with asymmetric cryptography, the message is encrypted using the recipients' public key. The recipient then decrypts the message using the private key. Because asymmetric ciphers tend to be computationally intensive (compared to symmetric encryption), they are usually used in combination with symmetric systems to implement public key cryptography. Asymmetric encryption is often used to transfer a session key rather than information proper - plaintext. This session key is then used to encrypt information using a symmetric encryption system. This gives the key exchange benefits of asymmetric encryption with the speed of symmetric encryption. A well known example of asymmetric encryption - also known as public key cryptography - is the RSA algorithm which is discussed later. This algorithm uses specific prime numbers (from which the private and public keys are composed) in order to realize the protocol.

In order to provide users with appropriate prime numbers, an infrastructure needs to be established by a third party whose 'business' is to distribute the public/private key pairs. This infrastructure is known as the Public Key Infrastructure or PKI. The use of a public key is convenient for those who wish to communicate with more than one individual and is thus a many-to-one protocol that avoids multiple key-exchange. On the other hand, a public key provides a basis for cryptanalysis. Given that $C = E_K(P)$ where K is the public key, the analyst can guess P and check the answer by comparing C with the intercepted ciphertext, a guess that is made easier if it is based on a known Crib - i.e. information that can be assumed to be a likely component of the plaintext. Public key algorithms are therefore often designed to resist chosen-plaintext attack. Nevertheless, analysis of public key and asymmetric systems in general reveals that the level of security is not as significant as that which can be achieved using a well-designed symmetric system. One obvious and fundamental issue relates to the third party responsible for the PKI and how much trust should be assumed, especially with regard to legislation concerning issues associated with the use of encrypted material.

C. Three-Way Pass Protocol

The three-way pass protocol, at first sight, provides a solution to the weaknesses associated with symmetric and

asymmetric encryption. Suppose that Alice writes a message, puts it in the box, locks the box with a lock whose combination number is known only to her and sends it onto Bob. Upon receipt Bob cannot open the box, so Bob locks the box with another lock whose combination number is known only to himself and sends it back to Alice. Upon receipt, Alice can remove her lock and send the box back to Bob (secured with his lock only) who is then able to remove his lock and recover the message. Note that by using this protocol, Alice and Bob do not need to agree upon a combination number; this avoids the weakness of symmetric encryption. Further, Alice and Bob do not need to send each other open locks which is a weakness of asymmetric encryption.

The problem with this protocol relates to the fact that it requires the message (secured in the locked box) to be exchanged three times. To explain this, suppose we have plaintext in the form of an American Standard Code for Information Interchange or ASCII-value array $p[i]$ say. Alice generates a cipher $n_1[i]$ using some appropriate strength random number generator and an initial condition based on some long integer - the key. Let the ciphertext $c[i]$ be generated by adding the cipher to the plaintext, i.e.

$$c_1[i] = p[i] + n_1[i]$$

which is transmitted to Bob. This is a substitution-based encryption process and is equivalent to Alice securing the message in the box with her lock - the first pass. Bob generates a new cipher $n_2[i]$ using the same (or possibly a different) random number generator with a different key and generates the ciphertext

$$c_2[i] = c_1[i] + n_2[i] = p[i] + n_1[i] + n_2[i]$$

which is transmitted back to Alice - the second pass. Alice now uses her cipher to generate

$$c_3[i] = c_2[i] - n_1[i] = p[i] + n_2[i]$$

which is equivalent to her taking off her lock from the box and sending the result back to Bob - the third pass. Bob then uses his cipher to recover the message, i.e.

$$c_3[i] - n_2[i] = p[i].$$

However, suppose that the cipher texts c_1, c_2 and c_3 are intercepted, then the plaintext array can be recovered since

$$p[i] = c_3[i] + c_1[i] - c_2[i].$$

This is the case for any encryption process that is commutative and associative. For example, if the arrays are considered to be bit streams and the encryption process undertaken using the XOR process (denoted by $\oplus$), then

$$\mathbf{c}_1 = \mathbf{n}_1 \oplus \mathbf{p},$$

$$\mathbf{c}_2 = \mathbf{n}_2 \oplus \mathbf{c}_1 = \mathbf{n}_2 \oplus \mathbf{n}_1 \oplus \mathbf{p},$$

$$\mathbf{c}_3 = \mathbf{n}_1 \oplus \mathbf{c}_2 = \mathbf{n}_2 \oplus \mathbf{p}$$

and

$$\mathbf{c}_1 \oplus \mathbf{c}_2 \oplus \mathbf{c}_3 = \mathbf{p}.$$

This is because for any bit stream $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$

$$\mathbf{a} \oplus \mathbf{a} \oplus \mathbf{b} = \mathbf{b}$$

and because the XOR operation is both commutative and associative i.e.

$$\mathbf{a} \oplus \mathbf{b} = \mathbf{b} \oplus \mathbf{a}$$

and

$$\mathbf{a} \oplus (\mathbf{b} \oplus \mathbf{c}) = (\mathbf{a} \oplus \mathbf{b}) \oplus \mathbf{c}.$$

These properties are equivalent to the fact that when Alice receives the box at the second pass with both locks on it, she can, in principle, remove the locks in any order. If, however, she had to remove Bob's lock before her own, then the protocol would become redundant.

D. Private Key Encryption

One of the principal goals in private key cryptography is to design Pseudo Random Number Generators (PRNGs) that provide outputs (random number streams) where no element can be predicted from the preceding elements given complete knowledge of the algorithm. Another important feature is to produce generators that have long cycle lengths. A further important feature, is to ensure that the Entropy of the random number sequence is a maximum, i.e. that the histogram of the number stream is uniform.

The use of modular integer arithmetic coupled with the use of prime numbers in the development of encryption algorithms tends to provide functions which are not invertible. They are one-way functions that can only be used to reproduce a specific (random) sequence of numbers from the same initial condition.

The basic idea in stream ciphers - as used for private key (symmetric) cryptography - is to convert a plaintext into a ciphertext using a key that is used as a seed for the PRNG. A plaintext file is converted to a stream of integer numbers using ASCII conversion. For example, suppose we wish to encrypt the authors surname *Blackledge* for which the ASCII[5] decimal integer stream or vector is

$$\mathbf{p} = (66, 108, 97, 99, 107, 108, 101, 100, 103, 101).$$

Suppose we now use the linear congruential PRNG defined by[6]

$$n_{i+1} = an_i \mathrm{mod} P$$

where $a = 13$, $P = 131$ and let the seed be 250659, i.e. $n_0 = 250659$. The output of this iteration is

$$\mathbf{n} = (73, 32, 23, 37, 88, 96, 69, 111, 2, 26).$$

If we now add the two vectors together, we generate the cipher stream

$$\mathbf{c} = \mathbf{n}+\mathbf{p} = (139, 140, 120, 136, 195, 204, 170, 211, 105, 127).$$

Clearly, provided the recipient of this number stream has access to the same algorithm (including the values of the parameters a and P) and crucially, to the same seed n_0,

[5]Any code can be used.

[6]Such a PRNG is not suitable for cryptography and is being used for illustrative purposes only.

the vector $\mathbf{n}$ can be regenerated and $\mathbf{p}$ obtained from $\mathbf{c}$ by subtracting $\mathbf{n}$ from $\mathbf{c}$. However, in most cryptographic systems, this process is usually accomplished using binary streams where the binary stream representation of the plaintext $\mathbf{p}$ and that of the random number stream or cipher $\mathbf{n}$ are used to generate the ciphertext binary stream $\mathbf{c}$ via the process

$$\mathbf{c} = \mathbf{n} \oplus \mathbf{p}.$$

Restoration of the plaintext is then accomplished via the operation

$$\mathbf{p} = \mathbf{n} \oplus \mathbf{c} = \mathbf{n} \oplus \mathbf{n} \oplus \mathbf{p}.$$

The processes discussed above are examples of digital confusion in which the information contained in the field $\mathbf{f}$ (the plaintext) is 'confused' using a stochastic function $\mathbf{c}$ (the cipher) via addition (decimal integer process) or with an XOR operator (binary process). Here, the seed plays the part of a key that it utilized for the process of encryption and decryption. This is an example of symmetric encryption in which the key is a private key known only to the sender and recipient of the encrypted message.

Given that the algorithm used to generate the random number stream has public access (together with the parameters it uses which are typically 'hard-wired' in order to provide a random field pattern with a long cycle length), the problem is how to securely exchange the key to the recipient of the encrypted message so that decryption can take place. If the key is particular to a specific communication and is used once and once only for this communication (other communications being encrypted using different keys), then the process is known as a one-time pad, because the key is only used once. Simple though it is, this process is not open to attack. In other words, no form of cryptanalysis will provide a way of deciphering the encrypted message. The problem is how to exchange the keys in a way that is secure and, thus, solutions to the key exchange problem are paramount in symmetric encryption,

The illustration of stream cipher encryption given above highlights the problem of key exchange, i.e. providing the value of n_0 to both sender and receiver. In addition to developing the technology for symmetric encryption (e.g. the algorithm or algorithms), it is imperative to develop appropriate protocols and procedures for using it effectively with the aim of reducing inevitable human error, the underlying principles being: (i) the elimination of any form of temporal correlation in the used algorithm; (ii) the generation of a key that is non-intuitive and at best random; (iii) the exchange of the key once it has been established.

E. Public-Private Key Encryption

Public-Private Key Encryption [36]-[40] is fundamentally asymmetric and in terms of the box and combination-lock paradigm is based on considering a lock which has two combinations, one to open the lock and another to lock it. The second constraint is the essential feature because one of the basic assumptions in the use of combination locks is that they can be locked irrespective of the rotor positions. Thus, after writing a message, Alice uses one of Bob's specially designed locks to lock the box using a combination number that is unique to Bob but is openly accessible to Alice and others who want to send Bob a message. This combination number is equivalent to the public key. Upon reception, Bob can open the lock using a combination number that is known only to himself - equivalent to a private key. However, to design such a lock, there must be some mechanical 'property' linking the combination numbers required to first lock it and then unlock it. It is this property that is the principal vulnerability associated with public/private key encryption, a property that is concerned with certain precise and exact relationships that are unique to the use of prime numbers and their applications with regard to generating pseudo random number streams and stochastic functions in general [41].

The most common example of a public-private key encryption algorithm is the RSA algorithm [39] which gets its name after the three inventors, Rivest, Shamir and Adleman who developed the generator in the mid 1970s[7]. It has since withstood years of extensive cryptanalysis. To date, cryptanalysis has neither proved nor disproved the security of the algorithm in a complete and self-consistent form which suggests a high confidence level in the algorithm.

The basic generator is given by

$$n_{i+1} = n_i^e \mathrm{mod}(pq)$$

where p, q and e are prime numbers and $e < pq$. Although this generator can be used to compute a pseudo random number stream n_i, the real value of the algorithm lies in its use for transforming plaintext P_i (taken to be a decimal integer array based on ASCII 7-bit code, for example) to ciphertext C_i directly via the equation

$$C_i = P_i^e \mathrm{mod}(pq).$$

We then consider the decryption process to be based on the same type of transform, i.e.

$$P_i = C_i^d \mathrm{mod}(pq).$$

The problem is then to find d given e, p and q. The 'key' to solving this problem is to note that if $ed - 1$ is divisible by $(p-1)(q-1)$, i.e. d is given by the solution of

$$de = \mathrm{mod}[(p-1)(q-1)]$$

then

$$C_i^d \mathrm{mod}(pq) = P_i^{ed} \mathrm{mod}(pq) = P_i \mathrm{mod}(pq)$$

using *Fermat's Little Theorem*, i.e. for any integer a and prime number p

$$a^p = a \mathrm{mod} p.$$

Note that this result is strictly dependent on the fact that $ed - 1$ is divisible by $(p-1)(q-1)$ making e a relative prime of $(p-1)(q-1)$ so that e and $(p-1)(q-1)$ have no common factors except for 1. This algorithm, is the basis for many public/private or asymmetric encryption methods. Here,

[7]There are some claims that the method was first developed at GCHQ in England and then re-invented (or otherwise) by Rivest, Shamir and Adleman in the USA; the method was not published openly by GCHQ - such are the realities of 'keeping it quiet'.

the public key is given by the number e and the product pq which are unique to a given recipient and in the public domain (like an individual's telephone number). Note that the prime numbers p and q and the number $e < pq$ must be distributed to Alice and Bob in such a way that they are unique to Alice and Bob on the condition that d exists! This requires an appropriate infrastructure to be established by a trusted third party whose 'business' is to distribute values of e, pq and d to its clients a Public Key Infrastructure. A PKI is required in order to distribute public keys, i.e. different but appropriate values of e and pq for use in public key cryptography (RSA algorithm). This requires the establishment of appropriate authorities and directory services for the generation, management and certification of public keys.

Recovering the plaintext from the public key and the cipher text can be conjectured to be equivalent to factoring the product of the two primes. The success of the system, which is one of the oldest and most popular public key cryptosystems, is based on the difficulty of factoring. The principal vulnerability of the RSA algorithm with regard to an attack is that e and pq are known and that p and q must be prime numbers - elements of a large but (assumed) known set. To attack the cipher, d must be found. But it is known that d is the solution of

$$de = \mathrm{mod}[(p-1)(q-1)]$$

which is only solvable if $e < pq$ is a relative prime of $(p-1)(q-1)$. An attack can therefore be launched by searching through prime numbers whose magnitudes are consistent with the product pq (which provides a search domain) until the relative prime condition is established for factors p and q. However, factoring pq to calculate d given e is not trivial. It is possible to attack an RSA cipher by guessing the value of $(p-1)(q-1)$ but this is no easier than factoring pq which is the most obvious means of attack. It is also possible for a cryptanalyst to try every possible d but this brute force approach is less efficient than trying to factor pq.

In general, RSA cryptanalysis (see Section IV) has shown that the attacks discovered to date illustrate the pitfalls to be avoided when implementing RSA. Thus, even though RSA ciphers can be attacked, the algorithm can still be considered secure when used properly. In order to ensure the continued strength of the cipher, RSA run factoring challenges on their websites. As with all PKI and other cryptographic products, this algorithm is possibly most vulnerable to authorities (at least those operating in the UK) having to conform to the Regulation of Investigatory Powers Act 2000, Section 49.

IV. Cryptanalysis

Any cryptographic system must be able to withstand cryptanalysis [42] and it is imperative that the design of any encryption method is subjected to different form of attack. Traditionally, the design of encryption systems is not always been based on the use of 'experts' with experience in the art of breaking ciphertext. Some of the ideas reported in this paper are based on experts with such experience and methods of cryptanalysis that depend critically on the encryption techniques which have been developed and, therefore, subject to (long) delays in publication!

Cryptanalysts work on 'attacks' to try and break a cryptosystem. In many cases, the cryptanalysts are aware of the algorithm used and will attempt to break the algorithm in order to compromise the keys or gain access to the actual plaintext. It is worth noting that even though a number of algorithms are freely published, this does not in any way mean that they are the most secure. Most government institutions and the military do not reveal the type of algorithm used in the design of a cryptosystem. The rationale for this is that, if we find it difficult to break a code with knowledge of the algorithm, then how much more difficult is it to break a code if the algorithm is unknown? On the other hand, within the academic community, security in terms of algorithm secrecy is not considered to be of high merit and publication of the algorithm(s) is always recommended. It remains to be understood whether this is a misconception within the academic world (due in part to the innocence associated with academic culture) or a covertly induced government policy (by those who are less innocent!). For example, in 2003, it was reported that the Americans had broken ciphers used by the Iranian intelligence services. What was not mentioned, was the fact that the Iranian ciphers were based on systems purchased indirectly from the USA and thus, based on USA designed algorithms [46].

The 'known algorithm' approach originally comes from the work of Auguste Kerchhoff. Kerchhoff's Principle states that: *A cryptosystem should be secure even if everything about the system, except the key, is public knowledge*. This principle was reformulated by Claude Shannon as *the enemy knows the system* and is widely embraced by cryptographers world wide. In accordance with the Kerchhoff-Shannon principle, the majority of civilian cryptosystems make use of publicly known algorithms. The principle is that of 'security through transparency' in which open-source software is considered to be inherently more secure than closed source software. On this basis there are several methods by which a system can be attacked where, in each case, it is assumed that the cryptanalyst has full knowledge of the algorithm(s) used.

A. Basic Attacks

We provide a brief overview of the basic attack strategies associated with cryptanalysis based on [44].

Ciphertext-only attack is where the cryptanalyst has a ciphertext of several messages at their disposal. All messages are assumed to have been encrypted using the same algorithm. The challenge for the cryptanalyst is to try and recover the plaintext from these messages. Clearly a cryptanalyst will be in a valuable position if they can recover the actual keys used for encryption.

Known-plaintext attack makes the task of the cryptanalysis simpler because, in this case, access is available to both the plaintext and the corresponding ciphertext. It is then necessary to deduce the key used for encrypting the messages, or design an algorithm to decrypt any new messages encrypted with the same key.

Chosen-plaintext attack involves the cryptanalyst possessing both the plaintext and the ciphertext. In addition, the analyst has the ability to encrypt plaintext and see the ciphertext produced. This provides a powerful tool from which the keys can be deduced.

Adaptive-chosen-plaintext attack is an improved version of the chosen-plaintext attack. In this version, the cryptanalyst has the ability to modify the results based on the previous encryption. This version allows the cryptanalyst to choose a smaller block for encryption.

Chosen-ciphertext attack can be applied when the cryptanalyst has access to several decrypted texts. In addition, the cryptanalyst is able to use the text and pass it though a 'black box' for an attempted decrypt. The cryptanalyst has to guess the keys in order to use this method which is performed on an iterative basis (for different keys), until a decrypt is obtained.

Chosen-key attack is based on some knowledge of the relationship between different keys and is not of practical significance except in special circumstances.

Rubber-hose cryptanalysis is based on the use of human factors such as blackmail and physical threat for example. It is often a very powerful attack and sometimes very effective.

Differential cryptanalysis is a more general form of cryptanalysis. It is the study of how differences in an input can affect differences in the output. This method of attack is usually based on a chosen plaintext, meaning that the attacker must be able to obtain encrypted ciphertexts for some set of plaintexts of their own choosing. This typically involves acquiring a Crib of some type as discussed in the following section.

Linear cryptanalysis is a known plaintext attack which uses linear relations between inputs and outputs of an encryption algorithm which holds with a certain probability. This approximation can be used to assign probabilities to the possible keys and locate the one that is most probable.

B. Cribs

The problem with any form of chosen-plaintext attack is, of course, how to obtain part or all of the plaintext in the first place. One method that can be used is to obtain a Crib. A Crib, a term that originated at Bletchley Park during the Second World War, is a plaintext which is known or suspected of being part of a ciphertext. If it is possible to compare part of the ciphertext that is known to correspond with the plaintext then, with the encryption algorithm known, one can attempt to identify which key has been used to generate the cipherext as a whole and thus decrypt an entire message. But how is it possible to obtain any plaintext on the assumption that all plaintexts are encrypted in their entirety? One way is to analyse whether or not there is any bad practice being undertaken by the user, e.g. sending stereotyped (encrypted) messages. Analysing any repetitive features that can be expected is another way of obtaining a Crib. For example, suppose that a user was writing letters using Microsoft word, for example, having established an electronic letter template with his/her name, address, company reference number etc. Suppose we assume that each time a new letter is written, the entire document is encrypted using a known algorithm. If it is possible to obtain the letter template then a Crib has been found. Assuming that the user is not prepared to share the electronic template (which would be a strange thing to ask for), a simple way of obtaining the Crib could be to write to the user in hardcopy and ask that the response from the same user is of the same type, pleading ignorance of all forms of ICT or some other excuse. This is typical of methods that are designed to seed a response that includes a useful Crib. Further, there are a number of passive cribs with regard to letter writing that can be assumed, the use of *Dear* and *Yours sincerely*, for example.

During the Second World War, when passive cribs such as daily weather reports became rare through improvements in the protocols associated with the use of Enigma and/or operators who took their work seriously, Bletchley Park would ask the Royal Air Force to create some 'trouble' that was of little military value. This included seeding a particular area in the North Sea with mines, dropping some bombs on the French coast or, for a more rapid response, asking fighter pilots to go over to France and 'shoot up' targets of opportunity[8], processes that came to be known as 'gardening'. The Enigma encrypted ciphertexts that were used to report the 'trouble' could then be assumed to contain information such as the name of the area where the mines had been dropped and/or the harbour(s) threatened by the mines. It is worth noting that the ability to obtain cribs by gardening was made relatively easy because of the war in which 'trouble' was to be expected and to be reported. Coupled with the efficiency of the German war machine with regard to its emphasis on accurate and timely reports, the British were in a privileged position in which they could create cribs at will and have some fun doing it!

When a captured and interrogated German stated that Enigma operators had been instructed to encode numbers by spelling them out, Alan Turing reviewed decrypted messages, and determined that the number 'eins' appeared in 90% of the messages. He automated the crib process, creating an 'Eins Catalogue', which assumed that 'eins' was encoded at all positions in the plaintext. The catalogue included every possible key setting which provided a very simple and effective way of recovering the key and is a good example of how the statistics (of a word or phrase) can be used in cryptanalysis.

The use of Enigma by the German naval forces (in particular, the U-boat fleet) was, compared to the German army and air force, made secure by using a password from one day to the next. This was based on a code book provided to the operator prior to departure from base. No transmission of the daily passwords was required, passive cribs were rare

[8]Using targets of opportunity became very popular towards the end of the war. Fighter pilots were encouraged to, in the words of General J Dolittle, 'get them in the air, get them on the ground, just get them'.

and seeding activities were difficult to arrange. Thus, if not for a lucky break, in which one of these code books (which were printed in ink that disappeared if they were dropped in seawater) was recovered intact by a British destroyer (HMS Bulldog) from a damaged U-boat (U-110) on May 9, 1941, breaking the Enigma naval transmissions under their time-variant code-book protocol would have been very difficult. A British Naval message dated May 10, 1941 reads: '1. Capture of U Boat 110 is to be referred to as operation Primrose; 2. Operation Primrose is to be treated with greatest secrecy and as few people allowed to know as possible...' Clearly, and for obvious reasons, the British were anxious to make sure that the Germans did not find out that U-110 and its codebooks had been captured and all the sailors who took part in the operation were sworn to secrecy. On HMS Bulldog's arrival back in Britain a representative from Bletchley Park, photographed every page of every book. The 'interesting piece of equipment' turned out to be an Enigma machine, and the books contained the Enigma codes being used by the German navy.

The U-boat losses that increased significantly through the decryption of U-boat Enigma ciphers led Admiral Carl Doenitz to suspect that his communications protocol had been compromised. He had no firm evidence, just a 'gut feeling' that something was wrong. His mistake was not to do anything about it[9], an attitude that was typical of the German High Command who were certifiable with regard to their confidence in the Enigma system. However, they were not uniquely certifiable. For example, on April 18, 1943, Admiral Yamamoto (the victor of Pearl Harbour) was killed when his plane was shot down while he was attempting to visit his forces in the Solomon Islands. Notification of his visit from Rabaul to the Solomon's was broadcast as Morse coded ciphertext over the radio, information that was being routinely decrypted by the Americans. At this point in the Pacific War, the Japanese were using a code book protocol similar to that used by the German Navy, in which the keys were changed on a daily basis, keys that the Americans had 'generated' copies of. Some weeks before his visit, Yamamoto had been given the option of ordering a new set of code books to be issued. He had refused to give the order on the grounds that the logistics associated with transferring new code books over Japanese held territory was incompatible with the time scale of his visit and the possible breach of security that could arise through a new code book being delivered into the hands of the enemy. This decision cost him his life. However, it is a decision that reflects the problems associated with the distribution of keys for symmetric cryptosystems especially when a multi-user protocol needs to be established for execution over a wide communications area. In light of this problem, Yamamoto's decision was entirely rational but, nevertheless, a decision based on the assumption that the cryptosystem had not already been compromised. Perhaps it was his 'faith in the system' and thereby his refusal to think the 'unthinkable' that cost him his life!

The principles associated with cryptanalysis that have been briefly introduced here illustrate the importance of using a dynamic approach to cryptology. Any feature of a security infrastructure that has any degree of consistency is vulnerable to attack. This can include plaintexts that have routine phrases such as those used in letters, the key(s) used to encrypt the plaintext and the algorithm(s) used for encryption. One of the principal advantages of using chaoticity for designing ciphers is that it provides the cryptographer with a limitless and dynamic resource for producing different encryption algorithms. These algorithms can be randomly selected and permuted to produce, in principle, an unlimited number of Meta encryption engines that operate on random length blocks of plaintext. The use of block cipher encryption is both necessary in order to accommodate the relatively low cycle length of chaotic ciphers and desirable in order to increase the strength of the cipher by implementing a multi-algorithmic approach. Whereas in conventional cryptography, emphasis focuses on the number of permutations associated with the keys used to 'seed' or 'drive' an algorithm, chaos-based encryption can focus on the number of permutations associated with the algorithms that are used, algorithms that can, with care and understanding, be quite literally 'invented on the fly'. Since cryptanalysis requires that the algorithm is known and concentrates on trying to find the key, this approach, coupled with other important details that are discussed later on in this paper, provides a method that can significantly enhance the cryptographic strength of the ciphertext. Further, in order to satisfy the 'innocence' of the academic community, it is, of course, possible to openly publish such algorithms (as in this paper, for example), but in the knowledge that many more can be invented and published or otherwise. This provides the potential for generating a host of 'home-spun' ciphers which can be designed and implemented by anyone who wishes to by-pass established practices and 'cook it themselves'.

V. Diffusion and Confusion

In terms of plaintexts, diffusion is concerned with the issue that, at least on a statistical basis, similar plaintexts should result in completely different ciphertexts even when encrypted with the same key. This requires that any element of the input block influences every element of the output block in an irregular fashion. In terms of a key, diffusion ensures that similar keys result in completely different ciphertexts even when used for encrypting the same block of plaintext. This requires that any element of the input should influence every element of the output in an irregular way. This property must also be valid for the decryption process because otherwise an intruder may be able to recover parts of the input from an observed output by a partly correct guess of the key used for encryption. The diffusion process is a function of the sensitivity to initial conditions, conditions that a cryptographic system should have and further, the inherent topological transitivity that the system should also exhibit causing the plaintext to be mixed through the action of the encryption process.

Confusion ensures that the (statistical) properties of plaintext blocks are not reflected in the corresponding ciphertext blocks. Instead every ciphertext must have a random ap-

[9]An instinct can be worth a thousand ciphers, ten-thousand if you like.

pearance to any observer and be quantifiable through appropriate statistical tests. However, diffusion and confusion are processes that are of fundamental importance in the design and analysis of cryptological systems, not only for the encryption of plaintexts but for data transformation in general.

A. The Diffusion Equation

In a variety of physical problems, the process of diffusion can be modelled in terms of certain solutions to the diffusion equation whose basic (linear) form is given by[10] [45]-[48]

$$\nabla^2 u(\mathbf{r},t) - \sigma \frac{\partial}{\partial t} u(\mathbf{r},t) = S(\mathbf{r},t),$$
$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2},$$
$$u(\mathbf{r},0) = u_0(\mathbf{r}), \quad \sigma = \frac{1}{D},$$

where D is the 'Diffusivity', S is a source term and u is a function which describes physical properties such as temperature, light, particle concentration and so on with initial value u_0.

The diffusion equation describes fields u that are the result of an ensemble of incoherent random walk processes, i.e. walks whose direction changes arbitrarily from one step to the next and where the most likely position after a time t is proportional to $\sqrt{t}$. Further, the diffusion equation differentiates between past and future, i.e. $\pm t$. This is because the diffusing field u represents the behaviour of some average property of an ensemble of many 'agents' which cannot in general go back to their original state. This fundamental property of diffusive processes has a synergy with the use of one-way functions in cryptology, i.e. functions that, given an input, produce an output that is not reversible - an output from which it is not possible to compute the input.

Consider the process of diffusion in which a source of material diffuses into a surrounding homogeneous medium, the material being described by some initial condition $u(\mathbf{r},0)$. Physically, it is to be expected that the material will increasingly 'spread out' as time evolves and that the concentration of the material decreases further away from the source. The general solution to the diffusion equation yields a result in which the spatial concentration of material is given by the convolution of the initial condition with a Gaussian function. This solution is determined by considering how the process of diffusion responds to a single point source which yields the Green's function (in this case, a Gaussian function) given by [48]-[50],

$$G(r,t) = \frac{1}{\sigma}\left(\frac{\sigma}{4\pi t}\right)^{\frac{n}{2}} \exp\left[-\left(\frac{\sigma r^2}{4t}\right)\right], t > 0, r = \mid \mathbf{r} \mid$$

which is the solution to

$$\left(\nabla^2 - \sigma\frac{\partial}{\partial t}\right) u(\mathbf{r},t) = \delta^n(\mathbf{r})\delta(t)$$

where δ denotes the Dirac delta function [51], [52] and $n = 1, 2, 3$ determines the dimension of the solution.

[10] $\mathbf{r} = \hat{\mathbf{x}}x + \hat{\mathbf{y}}y + \hat{\mathbf{z}}z$ denotes the spatial vector and t denotes time.

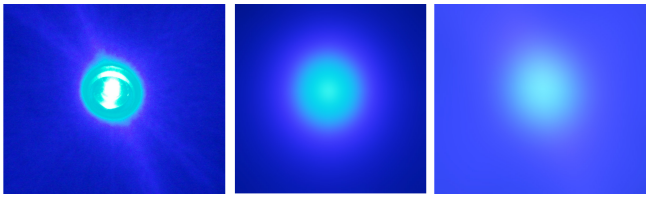


Fig. 2. Image of an optical source (left), the same source imaged through steam (centre) and a simulation of this effect obtained by convolving the source image with a Gaussian Point Spread Function (right).

In the infinite domain, the general solution to the diffusion equation can be written in the form [48]

$$u(\mathbf{r},t) = G(r,t) \otimes_{\mathbf{r}} \otimes_t S(\mathbf{r},t) + \sigma G(r,t) \otimes_{\mathbf{r}} u(\mathbf{r},0)$$

which requires that the spatial extent of the source function is infinite but can include functions that are localised provided that $S \to 0$ as $r \to \infty$ - a Gaussian function for example. The solution is composed of two terms. The first term is the convolution (in space and time, denoted by $\otimes_{\mathbf{r}}$ and $\otimes_t$ respectively) of the source function with the Green's function and the second term is the convolution (in space only) of the initial condition $u(\mathbf{r},0)$ with the same Green's function where

$$G(r,t) \otimes_{\mathbf{r}} \otimes_t S(\mathbf{r},t) = \int\int G(\mid \mathbf{r} - \mathbf{r}' \mid, t-\tau) S(\mathbf{r}',\tau) d^3\mathbf{r}' d\tau.$$

Thus, for example, in two-dimensions, for the case when $S = 0$, and ignoring scaling by $\sigma/(4\pi t)$), the solution for u is

$$u(x,y,t) = \exp\left[-\frac{\sigma}{4t}(x^2+y^2)\right] \otimes\otimes u_0(x,y)$$

where we have introduced $\otimes\otimes$ to denote the two-dimensional convolution integral. Here, the field at time $t > 0$ is given by the field at time $t = 0$ convolved with the two-dimensional Gaussian function $\exp[-\sigma(x^2 + y^2)/(4t)]$. This result can, for example, be used to model the diffusion of light through an optical diffuser. An example of such an effect is given in Figure 2 which shows a light source (the ceiling light of a steam room) imaged through air and through steam together with a simulation. Steam, as composed of a complex of small water droplets, effects light by scattering it a large number of times. The high degree of multiple scattering that takes place allows us to model the transmission of light in terms of a diffusive rather than a propagative process where the function u is taken to denote the intensity of light. The initial condition u_0 is taken to denote the initial image which is, in effect, the image of the light source recorded through air. As observed in Figure 2, the details associated with the light source are blurred through a mixing process which is determined by the Gaussian function that is characteristic of diffusion processes in general. In imaging science, functions of this type determine how a point of light is affected by the convolution process[11] and is thus referred to as the Point Spread Function or PSF [53]. The PSF is a particularly important characteristic of any imaging system in general, a characteristic that is related to the physical processes through which light is transformed from the *object plane* (input) to the *image plane* (output).

[11] Convolution is sometimes referred to by its German equivalent, i.e. by the word 'Faltung' which means 'mixing' or 'diffusing'.

If we record a diffused field u after some time $t = T$, is it possible to reconstruct the field at time $t = 0$, i.e. to solve the inverse problem or de-diffuse the field measured? We can express $u(\mathbf{r}, 0)$ in terms of $u(\mathbf{r}, T)$ using the Taylor series

$$u_0(\mathbf{r}) \equiv u(\mathbf{r}, 0) = u(\mathbf{r}, T) + \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} T^n \left[\frac{\partial^n}{\partial t^n} u(\mathbf{r}, t) \right]_{t=T}.$$

From the diffusion equation

$$\frac{\partial^2 u}{\partial t^2} = D\nabla^2 \frac{\partial u}{\partial t} = D^2 \nabla^4 u,$$
$$\frac{\partial^3 u}{\partial t^3} = D\nabla^2 \frac{\partial^2 u}{\partial t^2} = D^3 \nabla^6 u$$

and so on. Thus, in general we can write

$$\left[\frac{\partial^n}{\partial t^n} u(\mathbf{r}, t) \right]_{t=T} = D^n \nabla^{2n} u(x, y, T)$$

and after substituting this result into the series for u_0 given above, we obtain

$$u_0(\mathbf{r}) = u(\mathbf{r}, T) + \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} (DT)^n \nabla^{2n} u(\mathbf{r}, T)$$
$$\sim u - DT\nabla^2 u, \quad DT << 1.$$

B. Diffusion of a Stochastic Source

For the case when

$$\left(\nabla^2 - \sigma \frac{\partial}{\partial t} \right) u(\mathbf{r}, t) = -S(\mathbf{r}, t), \quad u(\mathbf{r}, 0) = 0$$

the solution is

$$u(\mathbf{r}, t) = G(r, t) \otimes_{\mathbf{r}} \otimes_t S(\mathbf{r}, t), \quad t > 0.$$

If a source is introduced in terms of an impulse at $t = 0$, then the 'system' will react accordingly and diffuse for $t > 0$. This is equivalent to introducing a source function of the form

$$S(\mathbf{r}, t) = s(\mathbf{r})\delta(t).$$

The solution is then given by

$$u(\mathbf{r}, t) = G(r, t) \otimes_{\mathbf{r}} s(\mathbf{r}), \quad t > 0.$$

Observe that this solution is of the same form as the homogeneous case with initial condition $u(\mathbf{r}, 0) = u_0(\mathbf{r})$ and that the solution for initial condition $u(\mathbf{r}, 0) = u_0(\mathbf{r})$ is given by

$$u(\mathbf{r}, t) = G(r, t) \otimes_{\mathbf{r}} [s(\mathbf{r}) + \sigma u_0(\mathbf{r})] = G(r, t) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n(\mathbf{r}, t)$$

where

$$n(\mathbf{r}, t) = \sigma G(r, t) \otimes_{\mathbf{r}} s(\mathbf{r}), \quad t > 0.$$

Note that if s is a stochastic function (i.e. a random dependent variable characterised, at least, by a Probability Density Function (PDF) denoted by $\Pr[s(\mathbf{r})]$), then n will also be a stochastic function. Also note that for a time-independent source function $S(\mathbf{r})$,

$$u_0(\mathbf{r}) = u(\mathbf{r}, T)$$
$$+ \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} [(DT)^n \nabla^{2n} u(\mathbf{r}, T) + D^{-1} \nabla^{2n-2} S(\mathbf{r})]$$

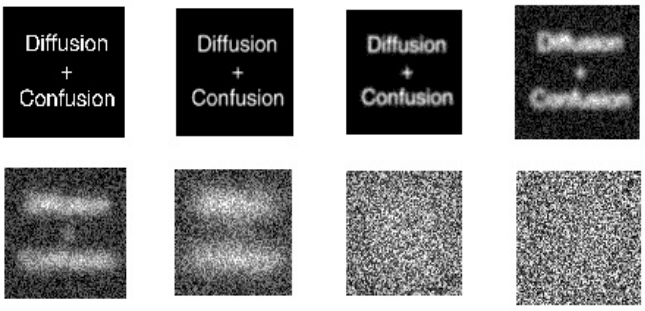


Fig. 3. Progressive diffusion and confusion of an image (top-left) - from left to right and from top to bottom - for uniform distributed noise. The convolution is undertaken using the convolution theorem and a Fast Fourier Transform (FFT)

and that if S is a stochastic function, then the field u can not be de-diffused (since it is not possible to evaluate u_0 exactly given $\Pr[S(\mathbf{r})]$). In other words, any error or 'noise' associated with diffusion leads to the process being irreversible - a 'one-way' process. This, however, depends on the magnitude of the diffusivity D which for large values cancels out the effect of any noise, thus making the process potentially reversible. In cryptography, it is therefore important that the process of diffusion applied (in order that a key affects every bit of the plaintext irrespective of the encryption algorithm that is used) has a low diffusivity.

The inclusion of a stochastic source function provides us with a self-consistent introduction to another important concept in cryptology, namely 'confusion'. Taking, for example, the two-dimensional case, the field u is given by

$$u(x, y) = \frac{\sigma}{4\pi t} \exp\left[-\frac{\sigma}{4t} (x^2 + y^2) \right] \otimes \otimes u_0(x, y) + n(x, y).$$

We thus arrive at a basic model for the process of diffusion and confusion, namely

Output=Diffusion+Confusion.

Here, *diffusion* involves the 'mixing' of the initial condition with a Gaussian function and *confusion* is compounded in the addition of a stochastic or noise function to the diffused output. The relative magnitudes of the two terms determines the dominating effect. As the noise function n increases in amplitude relative to the diffusion term, the output will become increasingly determined by the effect of confusion alone. In the equation above, this will occur as t increases since the magnitude of the diffusion term depends of the scaling factor $1/t$. This is illustrated in Figure 3 which shows the combined effect of diffusion and confusion for an image of the phrase

Confusion
+
Diffusion

as it is (from left to right and from top to bottom) progressively diffused (increasing values of t) and increasingly confused for a stochastic function n that is uniformly distributed. Clearly, the longer the time taken for the process of diffusion to occur, the more the output is confusion dominated. This is consistent

with all cases when the level of confusion is high and when the stochastic field used to generate this level of confusion is unknown (other than possible knowledge of its PDF). However, if the stochastic function has been synthesized[12] and is thus known *a priori*, then we can compute

$$u(x,y) - n(x,y) = \frac{\sigma}{4\pi t}\exp\left[-\frac{\sigma}{4t}(x^2+y^2)\right] \otimes\otimes u_0(x,y)$$

from which u_0 may be computed approximately via application of a deconvolution algorithm [53].

VI. Stochastic Fields

By considering the diffusion equation for a stochastic source, we have derived a basic model for the 'solution field' or 'output' $u(\mathbf{r},t)$ in terms of the initial condition or input $u_0(\mathbf{r})$. We now consider the principal properties of stochastic fields, considering the case where the fields are random variables that are functions of time t.

A. Independent Random Variables

Two random variables $f_1(t)$ and $f_2(t)$ are independent if their cross-correlation function is zero, i.e.

$$\int_{-\infty}^{\infty} f_1(t+\tau)f_2(\tau)d\tau = f_1(t) \odot f_2(t) = 0$$

where $\odot$ is used to denote the correlation integral above. From the correlation theorem [54], [55], it then follows that

$$F_1^*(\omega)F_2(\omega) = 0$$

where

$$F_1(\omega) = \int_{-\infty}^{\infty} f_1(t)\exp(-i\omega t)dt$$

and

$$F_2(\omega) = \int_{-\infty}^{\infty} f_2(t)\exp(-i\omega t)dt.$$

If each function has a PDF $\Pr[f_1(t)]$ and $\Pr[f_2(t)]$ respectively, the PDF of the function $f(t)$ that is the sum of $f_1(t)$ and $f_2(t)$ is given by the convolution of $\Pr[f_1(t)]$ and $\Pr[f_2(t)]$, i.e. the PDF of the function

$$f(t) = f_1(t) + f_2(t)$$

is given by [56], [57]

$$\Pr[f(t)] = \Pr[f_1(t)] \otimes \Pr[f_2(t)].$$

Further, for a number of statistically independent stochastic functions $f_1(t), f_2(t), ...$, each with a PDF $\Pr[f_1(t)], \Pr[f_2(t)], ...$, the PDF of the sum of these functions, i.e.

$$f(t) = f_1(t) + f_2(t) + f_3(t) + ...$$

is given by

$$\Pr[f(t)] = \Pr[f_1(t)] \otimes \Pr[f_2(t)] \otimes \Pr[f_1(t)] \otimes ...$$

[12]The synthesis of stochastic functions is a principal issue in cryptology.

These results can derived using the Characteristic Function [58]. For a strictly continuous random variable $f(t)$ with distribution function $P_f(x) = \Pr[f(t)]$ we define the expectation as

$$E(f) = \int_{-\infty}^{\infty} xP_f(x)dx,$$

which computes the mean value of the random variable, the Moment Generating Function as

$$E[\exp(-kf)] = \int_{-\infty}^{\infty} \exp(-kx)P_f(x)dx$$

which may not always exist and the Characteristic Function as

$$E[\exp(-ikf)] = \int_{-\infty}^{\infty} \exp(-ikx)P_f(x)dx$$

which will always exist. Observe that the moment generating function is the (two-sided) Laplace transform [59] of P_f and the Characteristic Function is the Fourier transform of P_f. Thus, if $f(t)$ is a stochastic function which is the sum of N independent random variables $f_1(t), f_2(t), ..., f_N(t)$ with distributions $P_{f_1}(x), P_{f_2}(x), ..., P_{f_N}(x)$, then

$$f(t) = f_1(t) + f_2(t) + ... + f_N(t)$$

and

$$E[\exp(-ikf)] = E[\exp[-ik(f_1 + f_2 + ... + f_N)]$$

$$= E[\exp(-ikf_1)]E[\exp(-ikf_2)]...E[\exp(-ikf_N)]$$

$$= \mathcal{F}_1[P_{f_1}]\mathcal{F}_1[P_{f_2}]...\mathcal{F}_1[P_{f_N}]$$

where $\mathcal{F}_1$ is the one-dimensional Fourier transform operartor defined as

$$\mathcal{F} \equiv \int_{-\infty}^{\infty} dx\exp(-ikx).$$

In other words, the Characteristic Function of the random variable $f(t)$ is the product of the Characteristic Functions for all random variables whose sum if $f(t)$. Using the convolution theorem for Fourier transforms, we then obtain

$$P_f(x) = \prod_{i=1}^{N}{}^{\otimes} P_{f_i}(x) = P_{f_1}(x) \otimes P_{f_2}(x) \otimes ... \otimes P_{f_N}(x).$$

Further, we note that if f_1, $f_2, ..., f_N$ are all identically distributed then

$$E[\exp[-ik(f_1 + f_2 + ... + f_N)] = (\mathcal{F}[P_{f_1}])^N$$

and

$$P_f(x) = P_{f_1}(x) \otimes P_{f_1}(x) \otimes ...$$

B. The Central Limit Theorem

The Central Limit Theorem stems from the result that the convolution of two functions generally yields a function which is smoother than either of the functions that are being convolved. Moreover, if the convolution operation is repeated, then the result starts to look more and more like a Gaussian function - a normal distribution - at least in an approximate sense [60], [61]. For example, suppose we have a number of independent random variables each of which is characterised by a distribution that is uniform. As we add more and more of these functions together, the resulting distribution is then given by convolving more and more of these (uniform) distributions. As the number of convolutions increases, the result tends to a Gaussian distribution. If we consider the effect of applying multiple convolutions of the uniform distribution

$$P(x) = \begin{cases} \frac{1}{X}, & \mid x \mid \leq X/2; \\ 0, & \text{otherwise} \end{cases}$$

then be considering the effect of multiple convolutions in Fourier space (through application of the convolution theorem) and working with a series representation of the result, it can be shown that (see Appendix I)

$$\prod_{i=1}^{N}{}^{\otimes} P_i(x) \equiv P_1(x) \otimes P_2(x) \otimes ... \otimes P_N(t)$$

$$\simeq \sqrt{\frac{6}{\pi N}} \exp(-6x^2/XN)$$

where $P_i(x) = P(x), \quad \forall i$ and N is large. Figure 4 illustrates the effect of successively adding uniformly distributed but independent random times series (each consisting of 500 elements) and plotting the resulting histograms (using 32 bins), i.e. given the discrete times series $f_1[i], f_2[i], f_3[i], f_4[i]$ for i=1 to 500, Figure 4 shows the time series

$$s_1[i] = f_1[i],$$

$$s_2[i] = f_1[i] + f_2[i],$$

$$s_3[i] = f_1[i] + f_2[i] + f_3[i],$$

$$s_4[i] = f_1[i] + f_2[i] + f_3[i] + f_4[i]$$

and the corresponding 32-bin histograms of the signals $s_j, j = 1, 2, 3, 4$. Clearly as j increases, the histogram starts to 'look' increasing normally distributed. Here, the uniformly distributed discrete time series $f_i, i = 1, 2, 3, 4$ have been computed using the uniform pseudo random number generator

$$f_{i+1} = af_i \mathrm{mod} P$$

where $a = 7^7$ and $P = 2^{32} - 1$ is a Mersenne prime number, by using four different seeds f_0 in order to provide time series that are 'independent'.

The Central Limit Theorem has been considered specifically for the case of uniformly distributed independent random variables. However, in general, it is approximately applicable for all independent random variables, irrespective of their

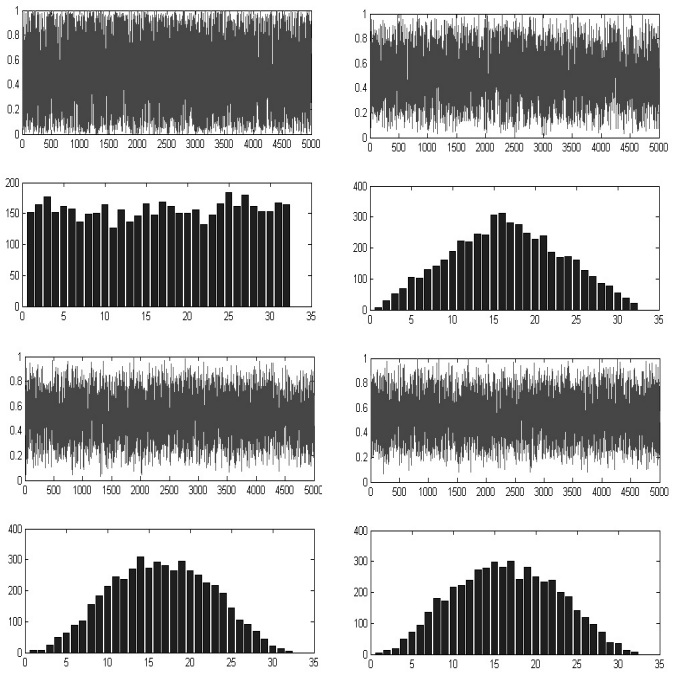


Fig. 4. Illustration of the Central Limit Theorem. The top-left image shows plots of a 500 element uniformly distributed time series and its histogram using 32 bins. The top-right image shows the result of adding two uniformly distributed and independent time series together and the 32 bin histogram. The bottom-left image is the result after adding three uniformly distributed times series and the bottom-right image is the result of adding four uniformly distributed times series.

distribution. In particular, we note that for a standard normal (Gaussian) distribution given by

$$\mathrm{Gauss}(x; \sigma, \mu) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

where

$$\int_{-\infty}^{\infty} \mathrm{Gauss}(x) dx = 1$$

and

$$\int_{-\infty}^{\infty} \mathrm{Gauss}(x) \exp(-ikx) dx = \exp(ik\mu) \exp\left(-\frac{\sigma^2 k^2}{2}\right),$$

then, since

$$\mathrm{Gauss}(x) \Longleftrightarrow \exp(ik\mu) \exp\left(-\frac{\sigma^2 k^2}{2}\right),$$

$$\prod_{j=1}^{N}{}^{\otimes} \mathrm{Gauss}(x) \Longleftrightarrow \exp(ikN\mu) \exp\left(-\frac{N\sigma^2 k^2}{2}\right)$$

so that

$$\prod_{j=1}^{N}{}^{\otimes} \mathrm{Gauss}(x) = \left(\frac{1}{2\pi N\sigma^2}\right) \exp\left[-\frac{1}{2N}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

where $\Longleftrightarrow$ denotes transformation form 'real' to 'Fourier space'. In other words, the addition of Gaussian distributed fields produces a Gaussian distributed field.

VII. Stochastic Diffusion

Given the classical diffusion/confusion model of the type

$$u(\mathbf{r}) = p(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n(\mathbf{r})$$

discussed above, we note that both the operator and the functional form of p are derived from solving a physical problem (using a Green's function solution) compounded in a particular PDE - diffusion equation. This is an example of 'Gaussian diffusion' since the characteristic Point Spread Function is a Gaussian function. However, we can use this basic model and consider a variety of PSFs as required. Although arbitrary changes to the PSF are inconsistent with classical diffusion, in cryptology we can, in principal, choose any PSF that is of value in 'diffusing' the data. For example, in Fresnel optics [62], [63], the PSF is of the same Gaussian form but with a complex exponential. If $f(x, y)$ is the 'object function' describing the 'object plane' and $u(x, y)$ is the image plane wave function, then [64], [65]

$$u(x, y) = p(x, y) \otimes \otimes f(x, y)$$

where the PSF p is given by (ignoring scaling) [53]

$$p(x, y) = \exp[i\alpha(x^2 + y^2)]; \quad \mid x \mid \leq X, \quad \mid y \mid \leq Y$$

where $\alpha = \pi/(z\lambda)$, λ being the wavelength and z the distance between the object and image planes, and where X and Y determine the spatial support of the PSF.

Stochastic diffusion involves interchanging the roles of p and n, i.e. replacing $p(\mathbf{r})$ - a deterministic PSF - with $n(\mathbf{r})$ - a stochastic function. Thus, noise diffusion is compounded in the result

$$u(\mathbf{r}) = n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + p(\mathbf{r})$$

where p can be any function or

$$u(\mathbf{r}) = n_1(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n_2(\mathbf{r})$$

where both n_1 and n_2 are stochastic function which may be of the same type (i.e. have the same PDFs) or of different types (with different PDFs). This form of diffusion is not 'physical' in the sense that it does not conform to a physical model as defined by the diffusion equation, for example. Here $n(\mathbf{r})$ can be any stochastic function (synthesized or otherwise).

The simplest form of noise diffusion is

$$u(\mathbf{r}) = n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}).$$

There are two approaches to solving the inverse problem: Given u and n, obtain u_0. We can invert or deconvolve by using the convolution theorem giving (for dimension $n = 1, 2, 3$)

$$u_0(\mathbf{r}) = \mathcal{F}_n^{-1}\left[\frac{U(\mathbf{k})N^*(\mathbf{k})}{\mid N(\mathbf{k}) \mid^2}\right]$$

where N is the Fourier transform of n and U is the Fourier transform of u. However, this approach requires regularisation in order to eliminate any singularities when $\mid N \mid^2 = 0$ through application of a constrained deconvolution filter such as the Wiener filter [53]. Alternatively, if n is the result of some random number generating algorithm, we can construct the stochastic field

$$m(\mathbf{r}) = \mathcal{F}_n^{-1}\left[\frac{N^*(\mathbf{k})}{\mid N(\mathbf{k}) \mid^2}\right]$$

where $\mid N(\mathbf{k}) \mid^2 > 0$, the diffused field now being given by

$$u(\mathbf{r}) = m(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}).$$

The inverse problem is then solved by correlating u with n, since

$$n(\mathbf{r}) \odot_{\mathbf{r}} u(\mathbf{r}) \Longleftrightarrow N^*(\mathbf{k})U(\mathbf{k})$$

and

$$N^*(\mathbf{k})U(\mathbf{k}) = N^*(\mathbf{k})M(\mathbf{k})U_0(\mathbf{k})$$

$$= N^*(\mathbf{k})\frac{N^*(\mathbf{k})}{\mid N(\mathbf{k}) \mid^2}U_0(\mathbf{k}) = U_0(\mathbf{k})$$

so that

$$u_0(\mathbf{r}) = n(\mathbf{r}) \odot_{\mathbf{r}} u(\mathbf{r}).$$

The condition that $\mid N(\mathbf{k}) \mid^2 > 0$ is simply achieved by implementing the following process: $\forall \mathbf{k}$, if $\mid N(\mathbf{k}) \mid^2 = 0$, then $\mid N(\mathbf{k}) \mid^2 = 1$. This result can be used to 'embed' one data field in another.

Consider the case when we have two independent images $I_1(x, y) \geq 0 \forall x, y$ and $I_2(x, y) \geq 0 \forall x, y$ and we consider the case of embedding I_1 with I_2. We construct a noise field $m(x, y) \geq 0 \forall x, y$ *a priori* and consider the equation

$$u(x, y) = Rm(x, y) \otimes \otimes I_1(x, y) + I_2(x, y)$$

where

$$\|m(x, y) \otimes \otimes I_1(x, y)\|_\infty = 1 \quad \text{and} \quad \|I_2(x, y)\|_\infty = 1.$$

By normalising the terms in this way, the coefficient $0 \leq R \leq 1$, can be used to adjust the relative magnitudes of the terms such that the diffused image I_1 is a perturbation of the 'host image' I_2. This provides us with a way of watermarking [66] one image with another, R being referred to as the watermarking ratio[13]. This approach could of course be implemented using a Fresnel diffuser. However, for applications in image watermarking, the diffusion of an image with a noise field provides a superior result because: (i) a noise field provides more uniform diffusion; (ii) noise fields can be generated using random number generators that depend on a single initial value or seed (i.e. a private key). An example of this approach is shown in Figure 5. Here an image I_2 (the 'host image') is watermarked by another image I_1 (the 'watermark image') and because $R = 0.1$, the output u is 'dominated' by the image I_2. The noise field n, is computed using a uniform random number generator in which the output array $\mathbf{n}$ is normalized so that $\|\mathbf{n}\|_\infty = 1$ and used to generate $n(x_i, y_i)$ on a row-by-row basis. Here, the seed is any integer such as $1873...$ which can be based on the application of a PIN (Personal Identity Number) or a password (e.g. 'Enigma', which in terms of an ASCII string - using binary to decimal conversion - is '216257556149'). Recovery of the watermark image requires

[13] Equivalent, in this application, to the standard term 'Signal-to-Noise' or SNR ratio as used in signal and image analysis.

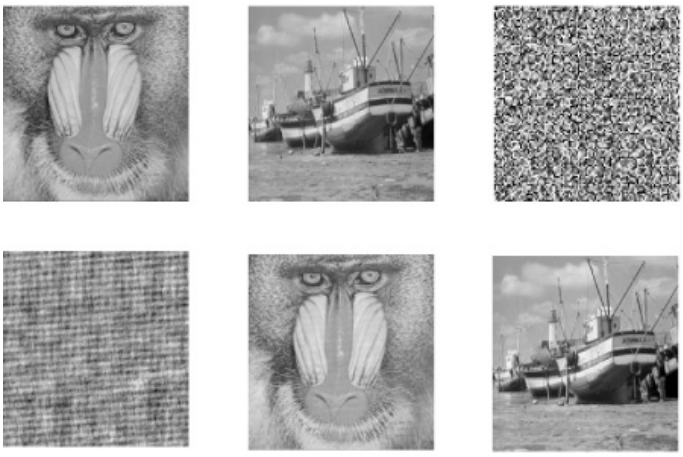


Fig. 5. Example of watermarking an image with another image using noise based diffusion. The 'host image' I_2 (top-left) is watermarked with the 'watermark image' I_1 (top-centre) using the diffuser (top-right) given by a uniform noise field n whose pixel-by-pixel values depend upon the seed used (the private key). The result of computing $m \otimes \otimes I_1$ (bottom-left) is added to the host image for $R = 0.1$ to generate the watermarked image u (bottom-centre). Recovery of the watermark image I_1 (bottom-right) is accomplished by subtracting the host image from the watermarked image and correlating the result with the noise field n.

knowledge of the PIN or Password and the host image I_2 The effect of adding the diffused watermark image to the host image yields a different, slightly brighter image because of the perturbation of I_2 by $Rm \otimes \otimes I_1$. This effect can be minimized by introducing a smaller watermarking ratio such that the perturbation is still recoverable by subtracting the host image from the watermarked image.

The expected statistical distribution associated with the output of a noise diffusion process is Gaussian. This can be shown if we consider u_0 to be a strictly deterministic function described by a sum of delta functions, equivalent to a binary stream in 1D or a binary image in 2D (discrete cases), for example. Thus if

$$u_0(\mathbf{r}) = \sum_i \delta^n(\mathbf{r} - \mathbf{r}_i)$$

then

$$u(\mathbf{r}) = n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) = \sum_{i=1}^{N} n(\mathbf{r} - \mathbf{r}_i).$$

Now, each function $n(\mathbf{r} - \mathbf{r}_i)$ is just $n(\mathbf{r})$ shifted by $\mathbf{r}_i$ and will thus be identically distributed. Hence

$$\Pr[u(\mathbf{r})] = \Pr\left[\sum_{i=1}^{N} n(\mathbf{r} - \mathbf{r}_i)\right] = \prod_{i=1}^{N}{}^{\otimes} \Pr[n(\mathbf{r})]$$

and from the Central Limit Theorem, we can expect $\Pr[u(\mathbf{r})]$ to be normally distributed for large N. This is illustrated in Figure 6 which shows the statistical distributions associated with a binary image, a uniformly distributed noise field and the output obtained by convolving the two fields together.

Given the equation

$$u(\mathbf{r}) = p(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n(\mathbf{r}),$$

if the diffusion by noise is based on interchanging p and n, then the diffusion of noise is based on interchanging u_0

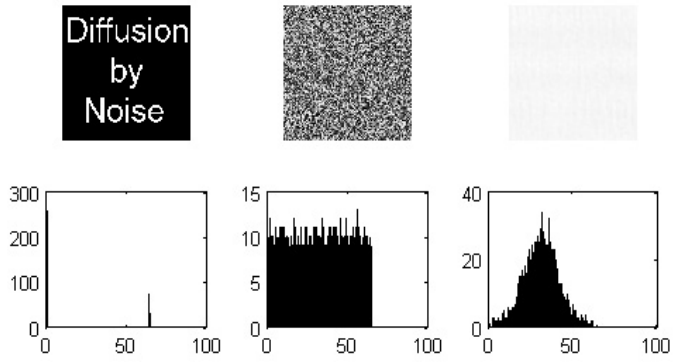


Fig. 6. Binary image (top-left), uniformly distributed 2D noise field (top-centre), convolution (top-right) and associated 64-bin histograms (bottom-left, -centre and -right respectively).

and n. In effect, this means that we consider the initial field u_0 to be a stochastic function. Note that the solution to the inhomogeneous diffusion equation for a stochastic source $S(\mathbf{r}, t) = s(\mathbf{r})\delta(t)$ is

$$n(\mathbf{r}, t) = G(r, t) \otimes_{\mathbf{r}} s(\mathbf{r})$$

and thus, n can be considered to be diffused noise. If we consider the model

$$u(\mathbf{r}) = p(\mathbf{r}) \otimes_{\mathbf{r}} n(\mathbf{r}),$$

then for the classical diffusion equation, the PSF is a Gaussian function. In general, given the convolution operation, p can be regarded as only one of a number of PSFs that can be considered in the 'production' of different stochastic fields u. This includes PSFs that define self-affine stochastic fields or random scaling fractals [67]-[69] that are based on fractional diffusion processes.

A. Print Authentication

The method discussed above refers to electronic-to-electronic type communications in which there is no loss of information. Steganography and watermarking techniques can be developed for hardcopy data which has a range of applications. These techniques have to be robust to the significant distortions generated by the printing and/or scanning process. A simple approach is to add information to a printed page that is difficult to see. For example, some modern colour laser printers, including those manufactured by HP and Xerox, print tiny yellow dots which are added to each page. The dots are barely visible and contain encoded printer serial numbers, date and time stamps. This facility provides a useful forensics tool for tracking the origins of a printed document which has only relatively recently been disclosed.

If the watermarked image is printed and scanned back into electronic form, then the print/scan process will yield an array of pixels that will be significantly different from the original electronic image even though it might 'look' the same. These differences can include the size of the image, its orientation, brightness, contrast and so on. Of all the processes involved in the recovery of the watermark, the subtraction of the host

image from the watermarked image is critical. If this process is not accurate on a pixel-by-pixel basis and deregistered for any of many reasons, then recovery of the watermark by correlation will not be effective. However, if we make use of the diffusion process alone, then the watermark can be recovered via a print/scan because of the compatibility of the processes involved. However, in this case, the 'watermark' is not covert but overt.

Depending on the printing process applied, a number of distortions will occur which diffuse the information being printed. Thus, in general, we can consider the printing process to introduce an effect that can be represented by the convolution equation

$$u_{\text{print}} = p_{\text{print}} \otimes \otimes u.$$

where u is the original electronic form of a diffused image (i.e. $u = n \otimes \otimes u_0$) and p_{print} is the point spread function of the printer. An incoherent image of the data, obtained using a flat bed scanner for example (or any other incoherent optical imaging system) will also have a characteristic point spread function p_{scan}. Thus, we can consider a scanned image to be given by

$$u_{\text{scan}} = p_{\text{scan}} \otimes \otimes u_{\text{print}}$$

where u_{scan} is taken to be the digital image obtained from the scan. Now, because convolution is commutative, we can write

$$u_{\text{scan}} = p_{\text{scan}} \otimes \otimes p_{\text{print}} \otimes \otimes p \otimes \otimes u_0 = p \otimes \otimes p_{\text{scan/print}} \otimes \otimes u_0$$

where

$$p_{\text{scan/print}} = p_{\text{scan}} \otimes \otimes p_{\text{print}}$$

which is the print/scan point spread function associated with the processing cycle of printing the image and then scanning it. By applying the method discussed earlier, we can obtain a reconstruction of the watermark whose fidelity is determined by the scan/print point spread function. However, in practice, the scanned image needs to be re-sized to that of the original. This is due to the scaling relationship (for a function f with Fourier transform F)

$$f(\alpha x, \beta y) \Longleftrightarrow \frac{1}{\alpha\beta} F\left(\frac{k_x}{\alpha}, \frac{k_y}{\beta}\right).$$

The size of any image captured by a scanner or other device will depend on the resolution used. The size of the image obtained will inevitably be different from the original because of the resolution and window size used to print the diffused image u and the resolution used to scan the image. Since scaling in the spatial domain causes inverse scaling in the Fourier domain, the scaling effect must be 'inverted' before the watermark can be recovered by correlation since correlation is not a scale invariant process. Re-sizing the image (using an appropriate interpolation scheme such as the bi-cubic method, for example) requires a set of two numbers n and m (i.e. the $n \times m$ array used to generate the noise field and execute the diffusion process) that, along with the seed required to regenerate the noise field, provides the 'private keys' needed to recover the data from the diffused image. An example of this approach is given in Figure 7 which shows the result of reconstructing four different images (a photograph, finger-print, signature and text) used in the design of an impersonalized debit/credit card. The use of 'diffusion only' watermarking for print security can be undertaken in colour by applying exactly the same diffusion/reconstruction methods to the red, green and blue components independently (as in Figure 7). This provides two additional advantages: (i) the effect of using colour tends to yield better quality reconstructions because of the colour combination process; (ii) for each colour component, it is possible to apply a noise field with a different seed. In this case, three keys are required to recover the watermark.

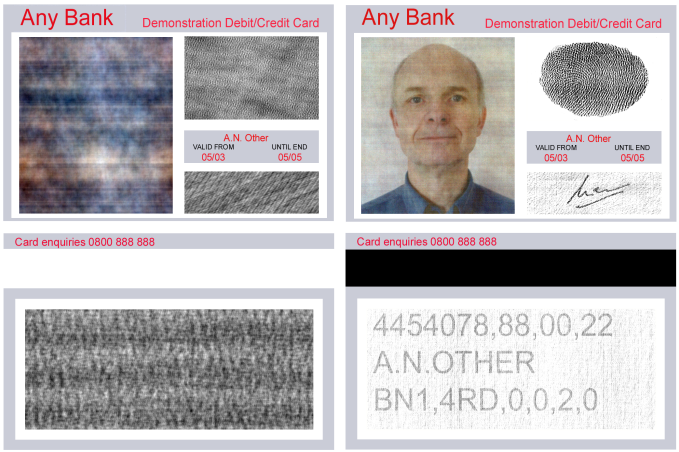


Fig. 7. Example of the application of 'diffusion only' watermarking. In this example, four images of a face, finger-print, signature and text have been diffused using the same noise field m and printed on the front (top-left) and back (bottom-left) of an impersonalized identity card using a 600 dpi printer. The reconstructions (top-right and bottom-right, respectively) are obtained using a conventional flat-bed scanner based on a 300 dpi grey-level scan.

Because this method is based on convolution alone and since

$$u_{\text{scan}} = p_{\text{scan/print}} \otimes \otimes u_0$$

as discussed earlier, the recovery of the f will not be negated by the distortion of the point spread function associated with the print/scan process, just limited or otherwise by its characteristics. Thus, if an image is obtained of the printed data field $p \otimes \otimes u_0$ which is out of focus due to the characteristics of $p_{\text{scan/print}}$, then the reconstruction of u_0 will be out of focus to the same degree. Decryption of images with this characteristic is only possible using an encryption scheme that is based on a 'diffusion only' approach. Figure 8 illustrates the recovery of a diffused image printed onto a personal identity card obtained using a flat bed scanner and then captured using mobile phone camera. In the latter case, the reconstruction is not in focus because of the wide-field nature of the lens used. However, the fact that recovery of the watermark is possible with a mobile phone means that the scrambled data can be transmitted securely and the card holders image (as in this example) recovered remotely and transmitted back to the same phone for authentication. This provides the necessary physical security needed to implement such a scheme in practice and means that specialist image capture devices are not required on site.

The diffusion process can be carried out using a variety

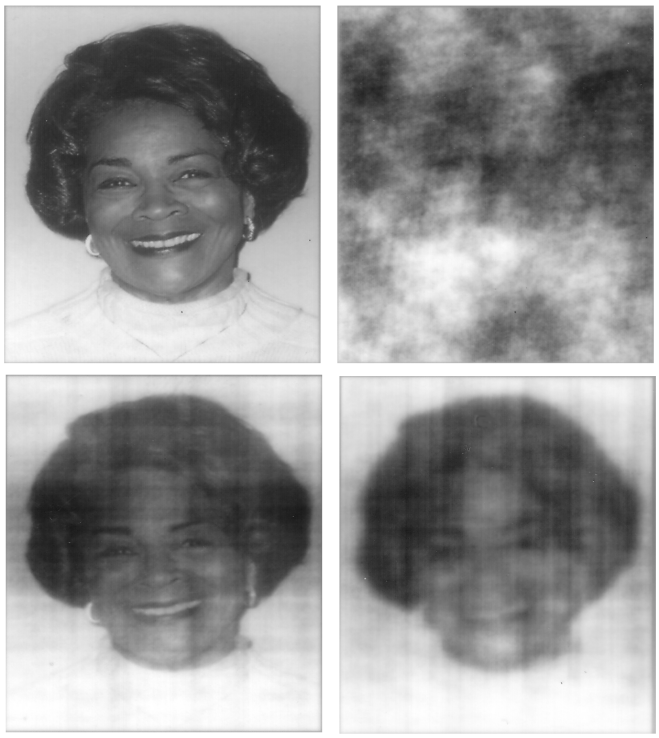


Fig. 8. Original image (top-left), diffused image (top-right), reconstruction using a flatbed scanner (bottom-left) and reconstruction using a mobile phone (bottom-right). These images have been scanned in grey scale from the original colour versions printed on to a personalised identity card at 600dpi stamp-size (i.e. 2cm×1.5cm).

of different noise fields other than the uniform noise field considered here. Changing the noise field can be of value in two respects: first, it allows a system to be designed that, in addition to specific keys, is based on specific algorithms which must be known *a priori*. These algorithms can be based on different pseudo uniform random number generators and/or different pseudo chaotic number generators that are post-processed to provide a uniform distribution of numbers. Second, the diffusion field depends on both the characteristics of the watermark image and the noise field. By utilizing different noise fields (e.g. Gaussian noise, Poisson noise, fractal noise and so on), the texture of the output field can be changed. The use of different noise fields is of value when different textures are required that are aesthetically pleasing and can be used to create a background that is printed over the entire document. In this sense, variable noise based diffusion fields can be used to replace complex print security features with the added advantage that, by de-diffusing them, information can be recovered. Further, these fields are very robust to data degradation created by soiling, for example. In the case of binary watermark images, data redundancy allows reconstructions to be generated from a binary output, i.e. after binarizing the diffusion field (with a threshold of 50% for example). This allows the output to be transmitted in a form that can tolerate low resolution and low contrast copying, e.g. a fax.

The tolerance of this method to printing and scanning is

Fig. 9. Example of the diffusion of composite images with the inclusion of a reference frame for enhancing and automating the processes of copping and orientation. In each case the data fields have been printed and scanned at 300 dpi.

excellent provided the output is cropped accurately (to within a few pixels) and oriented correctly. The processes of cropping and orientation can be enhanced and automated by providing a reference frame in which the diffused image is inserted. This is illustrated in Figure 9 which, in addition shows the effect of diffusing a combination of images. This has the effect of producing a diffused field that is very similar but nevertheless conveys entirely different information.

B. Covert Watermarking

Watermarking is usually considered to be a method in which the watermark is embedded into a host image in an unobtrusive way. Another approach is to consider the host image to be a data field that, when processed with another data field, generates new information.

Consider two images i_1 and i_2. Suppose we construct the following function

$$n = \mathcal{F}_2\left(\frac{I_1}{|\ I_1\ |^2} I_2\right)$$

where $I_1 = \mathcal{F}_2[i_1]$ and $I_2 = \mathcal{F}_2[i_2]$. If we now correlate n with i_1, then from the correlation theorem

$$i_1 \odot \odot n \Longleftrightarrow I_1^* \frac{I_1}{|\ I_1\ |^2} I_2 \Longleftrightarrow i_2.$$

In other words, we can recover i_2 from i_1 with a knowledge of n. Because this process is based on convolution and correlation alone, it is compatible and robust to printing and scanning, i.e. incoherent optical imaging. An example of this is given in Figure 10. In this scheme, the noise field n is the private key required to reconstruct the watermark and the host image can be considered to be a public key.

C. Application to Encryption

One of the principal components associated with the development of methods and algorithms to 'break' ciphertext is the analysis of the output generated by an attempted decrypt

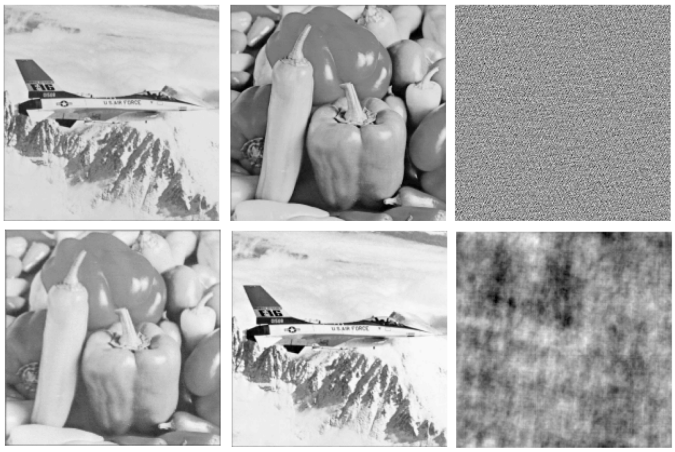


Fig. 10. Example of a covert watermarking scheme. i_1 (top-left) is 'processed' with i_2 (top-middle) to produce the noise field (top-right). i_2 is printed at 600 dpi, scanned at 300 dpi and then re-sampled back to its original size (bottom-left). Correlating this image with the noise field generates the reconstruction (bottom-centre). The reconstruction depends on just the host image and noise field. If the noise field and/or the host image are different or corrupted, then a reconstruction is not achieved, as in the example given (bottom-right).

and its evaluation in terms of an expected type. The output type is normally assumed to be plain text, i.e. the output is assumed to be in the form of characters, words and phrases associated with a natural language such as English or German, for example. If a plain text document is converted into an image file then the method described in the previous Section on 'covert watermarking' can be used to diffuse the plain text image i_2 using any other image i_1 to produce the field n. If both i_1 and n are then encrypted, any attack on these data will not be able to make use of an 'analysis cycle' which is based on the assumption that the decrypted output is plaintext. This approach provides the user with a relatively simple method of 'confusing' the cryptanalyst and invalidates attack strategies that have been designed and developed on the assumption that the encrypted data have been derived from plaintext alone.

VIII. Entropy Conscious Confusion and Diffusion

Consider a simple linear array such as a deck of eight cards which contains the ace of diamonds for example and where we are allowed to ask a series of sequential questions as to where in the array the card is. The first question we could ask is in which half of the array does the card occur which reduces the number of cards to four. The second question is in which half of the remaining four cards is the ace of diamonds to be found leaving just two cards and the final question is which card is it. Each successive question is the same but applied to successive subdivisions of the deck and in this way we obtain the result in three steps regardless of where the card happens to be in the deck. Each question is a binary choice and in this example, 3 is the minimum number of binary choices which represents the amount of information required to locate the card in a particular arrangement. This is the same as taking the binary logarithm of the number of possibilities, since $\log_2 8 = 3$. Another way of appreciating this result, is to consider a binary representation of the array of cards, i.e. 000,001,010,011,100,101,110,111, which requires three digits or bits to describe any one card. If the deck contained 16 cards, the information would be 4 bits and if it contained 32 cards, the information would be 5 bits and so on. Thus, in general, for any number of possibilities N, the information I for specifying a member in such a linear array, is given by

$$I = -\log_2 N = \log_2 \frac{1}{N}$$

where the negative sign is introduced to denote that information has to be acquired in order to make the correct choice, i.e. I is negative for all values of N larger than 1. We can now generalize further by considering the case where the number of choices N are subdivided into subsets of uniform size n_i. In this case, the information needed to specify the membership of a subset is given not by N but by N/n_i and hence, the information is given by

$$I_i = \log_2 P_i$$

where $P_i = n_i/N$ which is the proportion of the subsets. Finally, if we consider the most general case, where the subsets are non-uniform in size, then the information will no longer be the same for all subsets. In this case, we can consider the mean information given by

$$I = \sum_{i=1}^{N} P_i \log_2 P_i$$

which is the Shannon Entropy measure established in his classic works on information theory in the 1940s [70]. Information, as defined here, is a dimensionless quantity. However, its partner entity in physics has a dimension called 'Entropy' which was first introduced by Ludwig Boltzmann as a measure of the dispersal of energy, in a sense, a measure of disorder, just as information is a measure of order. In fact, Boltzmann's Entropy concept has the same mathematical roots as Shannon's information concept in terms of computing the probabilities of sorting objects into bins (a set of N into subsets of size n_i) and in statistical mechanics the Entropy is defined as [71], [72]

$$E = -k \sum_i P_i \ln P_i$$

where k is Boltzmann's constant. Shannon's and Boltzmann's equations are similar. E and I have opposite signs, but otherwise differ only by their scaling factors and they convert to one another by $E = -(k \ln 2)I$. Thus, an Entropy unit is equal to $-k \ln 2$ of a bit. In Boltzmann's equation, the probabilities P_i refer to internal energy levels. In Shannon's equations P_i are not *a priori* assigned such specific roles and the expression can be applied to any physical system to provide a measure of order. Thus, information becomes a concept equivalent to Entropy and any system can be described in terms of one or the other. An increase in Entropy implies a decrease of information and vise versa. This gives rise to the fundamental conservation law: *The sum of (macroscopic) information change and Entropy change in a given system is zero.*

From the point of view of designing an appropriate substitution cipher, the discussion above clearly dictates that the cipher

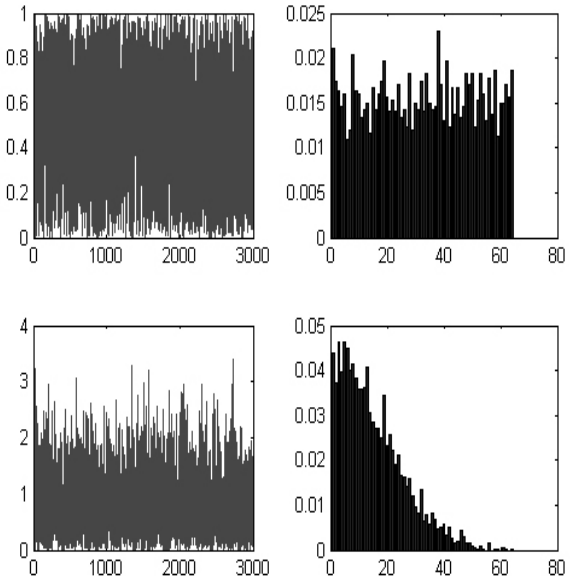


Fig. 11. A 3000 element uniformly distributed random number stream (top left) and its 64-bin discrete PDF (top right) with $I = 4.1825$ and a 3000 element Gaussian distributed random number stream (bottom left) and its 64-bin discrete PDF (bottom right) with $I = 3.2678$.

$n[i]$ should be such that the Entropy of the ciphertext $u[i]$ is a maximum. This requires that a PRNG algorithm be designed that outputs a number stream whose Entropy is maximum - as large as is possible in practice. The stream should have a PDF P_i that yields the largest possible values for I. Figure 11 shows a uniformly distributed and a Gaussian distributed random number stream consisting of 3000 elements and the characteristic discrete PDFs using 64-bins (i.e. for $N = 64$). The Information Entropy, which is computed directly from the PDFs using the expression for I given above, is always greater for the uniformly distributed field. This is to be expected because, for a uniformly distributed field, there is no bias associated with any particular numerical range and hence, no likelihood can be associated with a particular state. Hence, one of the underlying principles associated with the design of a cipher $n[i]$ is that it should output a uniformly distributed sequence of random numbers. However, this does not mean that the ciphertext itself will be uniformly distributed since if

$$u(\mathbf{r}) = u_0(\mathbf{r}) + n(\mathbf{r})$$

then

$$\Pr[u(\mathbf{r})] = \Pr[u_0(\mathbf{r})] \otimes \Pr[n(\mathbf{r})].$$

This is illustrated in Figure 12 which shows 128-bin histograms for a 7-bit ASCII plaintext (the LaTeX file associated with this paper) $u_0[i]$, a stream of uniformly distributed integers $n[i]$, $0 \leq n \leq 127$ and the ciphertext $u[i] = u_0[i] + n[i]$. The spike associate with the plaintext histogram reflects the 'character' that is most likely to occur in the plaintext of a natural Indo-European language, i.e. a space with ASCII value 32. Although the distribution of the ciphertext is broader than the plaintext it is not as broad as the cipher and certainly not uniform. Thus, the Entropy of the ciphertext, although larger than the plaintext (in this example $I_{u_0} = 3.4491$ and

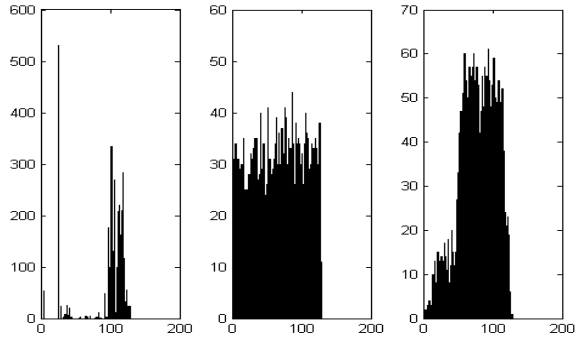


Fig. 12. 128-bin histograms for an 7-bit ASCII plaintext $u_0[i]$ (left), a stream of uniformly distributed integers between 0 and 127 $n[i]$ (centre) and the substitution cipher $u[i]$ (right).

$I_u = 5.3200$), the Entropy of the ciphertext is still less than that of the cipher (in this example $I_n = 5.5302$). There are two ways in which this problem can be solved. The first method is to construct a cipher n with a PDF such that

$$P_n(x) \otimes P_{u_0}(x) = U(x)$$

where $U(x) = 1, \quad \forall x$. Then

$$P_n(x) = U(x) \otimes Q(x)$$

where

$$Q(x) = \mathcal{F}_1^{-1}\left(\frac{1}{\mathcal{F}[P_{u_0}(x)]}\right).$$

But this requires that the cipher is generated in such a way that its output conforms to an arbitrary PDF as determined by the plaintext to be encrypted. The second method is based on assuming that the PDF of all plaintexts will be of the form given in Figure 12 with a characteristic dominant spike associated with the number of spaces that occur in the plaintext[14]. Noting that

$$P_n(x) \otimes \delta(x) = P_n(x)$$

then as the amplitude of the spike increases, the output increasingly approximates a uniform distribution; the Entropy of the ciphertext increases as the Entropy of the plaintext decreases. One simple way to implement this result is to pad-out the plaintext with a single character. Padding out a plaintext file with any character provides a ciphertext with a broader distribution, the character ? (with an ASCII decimal integer of 63) providing a symmetric result. The statistical effect of this is illustrated in Figure 13 where $I_{u_0} = 1.1615, \quad I_n = 5.5308$ and $I_u = 5.2537$.

IX. Statistical Properties of a Cipher

Diffusion has been considered via the properties associated with the homogeneous (classical) diffusion equation and the general Green's function solution. Confusion has been considered through the application of the inhomogeneous diffusion

[14]This is only possible provided the plaintext is an Indo-European alphanumeric array and is not some other language or file format - a compressed image file, for example.

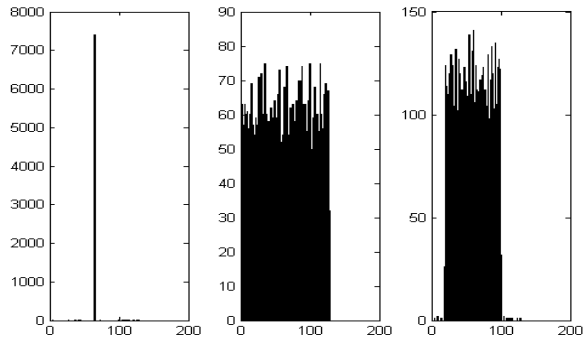


Fig. 13. 127-bin histograms for an 7-bit ASCII plaintext $u_0[i]$ (left) after space-character padding, a stream of uniformly distributed integers between 0 and 255 $n[i]$ (centre) and the substitution cipher $u[i]$ (right).

equation with a stochastic source function and it has been shown that

$$u(\mathbf{r}) = p(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n(\mathbf{r})$$

where p is a Gaussian Point Spread Function and n is a stochastic function.

Diffusion of noise involves the case when u_0 is a stochastic function. Diffusion by noise involves the use of a PSF p that is a stochastic function. If u_0 is taken to be deterministic information, then we can consider the processes of noise diffusion and confusion to be compounded in terms of the following:

Diffusion

$$u(\mathbf{r}) = n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}).$$

Confusion

$$u(\mathbf{r}) = u_0(\mathbf{r}) + n(\mathbf{r}).$$

Diffusion and Confusion

$$u(\mathbf{r}) = n_1(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r}) + n_2(\mathbf{r}).$$

The principal effects of diffusion and confusion have been illustrated using various test images. This has been undertaken for visual purposes only but on the understanding that such 'effects' apply to fields in different dimensions in a similar way.

The statistical properties associated with independent random variables has also been considered. One of the most significant results associated with random variable theory is compounded in the Central Limit Theorem. When data is recorded, the stochastic term n, is often the result of many independent sources of noise due to a variety of physical, electronic and measuring errors. Each of these sources may have a well-defined PDF but if n is the result of the addition of each of them, then the PDF of n tends to be Gaussian distributed. Thus, Gaussian distributed noise tends to be common in the large majority of applications in which u is a record of a physical quantity.

In cryptology, the diffusion/confusion model is used in a variety of applications that are based on diffusion only, confusion only and combined diffusion/confusion models. One such example of the combined model is illustrated in Figure 5 which shows how one data field can be embedded in another field (i.e. how one image can be used to watermark another image using noise diffusion). In standard cryptography, one of the most conventional methods of encrypting information is through application of a confusion only model. This is equivalent to implementing a model where it is assumed that the PSF is a delta function so that

$$u(\mathbf{r}) = u_0(\mathbf{r}) + n(\mathbf{r}).$$

If we consider the discrete case in one-dimension, then

$$u[i] = u_0[i] + n[i]$$

where $u_0[i]$ is the plaintext array or just 'plaintext' (a stream of integer numbers, each element representing a symbol associated with some natural language, for example), $n[i]$ is the 'cipher' and $u[i]$ is the 'ciphertext'. Methods are then considered for the generation of stochastic functions $n[i]$ that are best suited for the generation of the ciphertext. This is the basis for the majority of substitution ciphers where each value of each element of $u_0[i]$ is substituted for another value through the addition of a stochastic function $n[i]$, a function that should: (i) include outputs that are zero in order that the spectrum of random numbers is complete[15]; (ii) have a uniform PDF. The conventional approach to doing this is to design appropriate PRNGs or, as discussed later in this work, pseudo chaotic ciphers. In either case, a cipher should be generated with maximum Entropy which is equivalent to ensuring that the cipher is a uniformly distributed stochastic field. However, it is important to appreciate that the statistics of a plaintext are not the same as those of the cipher when encryption is undertaken using a confusion only model; instead the statistics are determined by the convolution of the PDF of the plaintext with the PDF of the cipher. Thus, if

$$u(\mathbf{r}) = u_0(\mathbf{r}) + n(\mathbf{r})$$

then

$$\Pr[u(\mathbf{r})] = \Pr[n(\mathbf{r})] \otimes_r \Pr[u_0(\mathbf{r})].$$

One way of maximising the Entropy of u is to construct u_0 such that $\Pr[u_0(\mathbf{r})] = \delta(\mathbf{r})$. A simple and practical method of doing this is to pad the data u_0 with a single element that increases the data size but does not intrude on the legibility of the plaintext.

Assuming that the encryption of a plaintext u_0 is undertaken using a confusion only model, there exists the possibility of encrypting the ciphertext again. This is an example of double encryption, a process that can be repeated an arbitrary number of times to give triple and quadruple encrypted outputs. However, multiple encryption procedures in which

$$u(\mathbf{r}) = u_0(\mathbf{r}) + n_1(\mathbf{r}) + n_2(\mathbf{r}) + ...$$

[15]The Enigma cipher, for example, suffered from a design fault with regard to this issue in that a letter could not reproduce its self - $u[i] \neq u_0[i] \forall i$. This provided a small statistical bias which was nevertheless significant in the decryption of Enigma ciphers.

where n_1, n_2,... are different ciphers, each consisting of uniformly distributed noise, suffer from the fact that the resultant cipher is normally distributed because, from the Central Limit Theorem

$$\Pr[n_1 + n_2 + ...] \sim \text{Gauss}(x).$$

For this reason, multiple encryption systems are generally not preferable to single encryption systems. A notable example is the triple DES (Data Encryption Standard) or DES3 system [73] that is based on a form of triple encryption and originally introduced to increase the key length associated with the generation of a single cipher n_1. DES3 was endorsed by the National Institute of Standards and Technology (NIST) as a temporary standard to be used until the Advanced Encryption Standard (AES) was completed in 2001 [74].

The statistics of an encrypted field formed by the diffusion of u_0 (assumed to be a binary field) with noise produces an output that is Gaussian distributed, i.e. if

$$u(\mathbf{r}) = n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r})$$

then

$$\Pr[u(\mathbf{r})] = \Pr[n(\mathbf{r}) \otimes_{\mathbf{r}} u_0(\mathbf{r})] \sim \text{Gauss}(x).$$

Thus, the diffusion of u_0 produces an output whose statistics are not uniform but normally distributed. The Entropy of a diffused field using uniformly distributed noise is therefore less than the Entropy of a confused field. It is for this reason, that a process of diffusion should ideally be accompanied by a process of confusion when such processes are applied to cryptology in general.

The application of noise diffusion for embedding or watermarking one information field in another is an approach that has a range of applications in covert ciphertext transmission. However, since the diffusion of noise by a deterministic PSF produces an output whose statistics tend to be normally distributed, such fields are not best suited for encryption. However, this process is important in the design of stochastic fields that have important properties for the camouflage of encrypted data.

X. Iterated Function Systems and Chaos

In cryptography, the design of specialized random number generators with idealized properties forms the basis of many of the algorithms that are applied. Although the type of random number generators considered so far are of value in the generation of noise fields, the properties of these algorithms are not well suited for cryptography especially if the cryptosystem is based on a public domain algorithm. This is because it is relatively easy to apply brute force attacks in order to recover the parameters used to 'drive' a known algorithm especially when there is a known set of rules required to optimise the algorithm in terms of parameter specifications. In general stream ciphers typically use an iteration of the type

$$x_{i+1} = f(x_i, p_1, p_2, ...)$$

where p_i is some parameter set (e.g. prime numbers) and x_0 is the key. The cipher $\mathbf{x}$, which is usually of decimal integer type, is then written in binary form (typically using ASCII 7-bit code) and the resulting bit stream used to encrypt the plaintext (after conversion to a bit stream with the same code) using an XOR operation. The output bit stream can then be converted back to ASCII ciphertext form as required. Decryption is then undertaken by generating the same cipher (for the same key) and applying an XOR operation to the ciphertext (binary stream). The encryption/decryption procedure is thus of the same type and attention is focused on the characteristics of the algorithm that is used for computing the cipher. However, whatever algorithm is designed and irrespective of its 'strength' and the length of the key that is used, in all cases, symmetric systems require the users to exchange the key. This requires the use of certain *key exchange algorithms*. Stream ciphers are essential Vernam type ciphers which encrypt bit streams on a bit by bit basis. By comparison block ciphers operate of blocks of the stream and may apply permutations and shifts to the data which depend on the key used. In this section we provide the foundations for the use of IFS for generating Vernam ciphers that are constructed from random length blocks of data that are based on the application different IFS.

A. Background to Chaos

The word Chaos appeared in early Greek writings and denoted either the primeval emptiness of the universe before things came into being or the abyss of the underworld [11]. Both concepts occur in the Theogony of Hesiod[16]. This concept tied in with other early notions that saw in Chaos the darkness of the underworld. In later Greek works, Chaos was taken to describe the original state of things, irrespective of the way they were conceived. The modern meaning of the word is derived from Ovid (Publius Ovidius Naso - known to the English speaking world as *Ovid*), a Roman poet (43BC - 17AD) and a major influence in early Latin literature, who saw Chaos as the original disordered and formless mass, from which the ordered universe was derived.

The modern notion of chaos - apart from being a term to describe a mess - is connected with the behaviour of dynamical systems that appear to exhibit erratic and non-predictable behaviour but, on closer 'inspection', reveal properties that have definable 'structures'. Thus, compared with the original Greek concept of chaos, chaotic systems can reveal order, bounded forms and determinism, a principal feature being their self-organisation and characterisation in terms of self-affine structures. This aspect of chaos immediately suggests that chaotic systems are not suitable for applications to cryptography which requires ciphers that have no predictable dynamic behaviour or structure of any type, e.g. pseudo random number streams that are uniformly distributed with maximum entropy. However, by applying appropriate post-conditioning criterion to a pseudo chaotic number stream, a cipher can be designed that has the desired properties.

The idea that a simple nonlinear but entirely deterministic systems can behave in an apparently unpredictable and chaotic manner was first noticed by the great French mathematician

[16]Hesiod, 700 BC, one of the earliest Greek poets. His epic 'Theogony' describes the myths of the gods.

Henri Poincaré in the late Nineteenth Century. In spite of this, the importance of chaos was not fully appreciated until the widespread availability of digital computers for numerical simulations and the demonstration of chaos in various physical systems. In the early 1960s, the American mathematician, Edward Lorenz re-discovered Poincaré's observations while investigating the numerical solution of a system of non-linear equations used to model atmospheric turbulence, equations that are now known as the Lorenz equations.

A primary feature of chaotic systems is that they exhibit self-affine structures when visualised and analysed in an appropriate way, i.e. an appropriate phase space. In this sense, the geometry of a chaotic system may be considered to be fractal. This is the principal feature that provides a link between chaotic dynamics and fractal geometry.

A key feature of chaotic behaviour in different systems is the sensitivity to initial conditions. Thus, *'It may happen that small differences in the initial conditions produce very great ones in the final phenomena. A small error in the former will produce an enormous error in the future. Prediction becomes impossible'* (Edward Lorenz[17]). This aspect of a chaotic system is ideal for encryption in terms of the diffusion requirement discussed earlier, i.e. that a cryptographic system should be highly sensitive to the initial conditions (the key) that is applied. However, in a more general context, the sensitivity to initial conditions of chaotic systems theory is an important aspect of using the theory to develop a mathematical description of complex phenomena such a Brownian and fractional Brownian process, weather changes in meteorology or population fluctuations in biology. The relative success of chaos theory for modelling complex phenomena has caused an important paradigm shift that has provided the first 'scientific' explanation for the coexistence of such concepts as law and disorder, determinism and unpredictability.

Formally, chaos theory can be defined as the study of complex nonlinear dynamic systems. The word 'complex' is related to the recursive and nonlinear characteristics of the algorithms involved and the word 'dynamic' implies the non-constant and non-periodic nature of such systems. Chaotic systems are commonly based on recursive processes, either in the form of single or coupled algebraic equations or a set of (single or coupled) differential equations modelling a physical or virtual system.

Chaos is often but incorrectly associated with noise in that it is taken to represent a field which is unpredictable. Although this is the case, a field generated by a chaotic system generally has more structure if analysed in an appropriate way, a 'structure' that may exhibit features that are similar at different scales. Thus, chaotic fields are not the same as noise fields either in terms of their behaviour or the way in which they are generated. Simple chaotic fields are typically the product of an iteration of the form $x_{i+1} = f(x_i)$ where the function f is some nonlinear map which depends on a single or a set of parameters. The chaotic behaviour of x_i depends critically of the value of the parameter(s). The iteration process may not necessarily be a single nonlinear mapping but consist of a set of nonlinear coupled equations of the form

$$\begin{aligned} x_{i+1}^{(1)} &= f_1(x_i^{(1)}, x_i^{(2)}, ..., x_i^{(N)}), \\ x_{i+1}^{(2)} &= f_2(x_i^{(1)}, x_i^{(2)}, ..., x_i^{(N)}), \\ &\vdots \\ x_{i+1}^{(N)} &= f_N(x_i^{(1)}, x_i^{(2)}, ..., x_i^{(N)}) \end{aligned}$$

where the functions $f_1, f_2, ..., f_N$ may all be nonlinear or nonlinear and linear. In turn, such a coupled system can be the result of many different physical models covering a wide range of applications in science and engineering.

B. Vurhulst Processes and the Logistic Map

Suppose there is a fixed population of N individuals living on an island (with no one leaving or entering) and a fatal disease (for which there is no cure) is introduced, which is spread through personal contact causing an epidemic to break out. The rate of growth of the disease will normally be proportional to the number of carriers c say. Suppose we let $x = c/N$ be the proportion of individuals with the disease so that $100x$ is the percentage of the population with the disease. Then, the equation describing the rate of growth of the disease is

$$\frac{dx}{dt} = kx$$

whose solution is

$$x(t) = x_0 \exp(kt)$$

where x_0 is the proportion of the population carrying the disease at $t = 0$ (i.e. when the disease first 'strikes') and k is a constant of proportionality defining the growth rate. The problem with this conventional growth rate model is that when $x = 1$, there can be no further growth of the disease because the island population no longer exists and so we must impose the condition that $0 < x(t) \leq 1, \ \forall t$. Alternatively, suppose we include the fact that the rate of growth must also be proportional to the number of individuals $1 - x$ who do not become carriers, due to isolation of their activities and/or genetic disposition, for example. Then, our rate equation becomes

$$\frac{dx}{dt} = kx(1 - x)$$

and if $x = 1$, the epidemic is extinguished. This equation can be used to model a range of situations similar to that introduced above associated with predator-prey type processes. (In the example given above, the prey is the human and the predator could be a virus or bacterium, for example). Finite differencing over a time interval Δt, we have

$$\frac{x_{i+1} - x_i}{\Delta t} = kx_i(1 - x_i)$$

or

$$x_{i+1} = x_i + k\Delta t x_i(1 - x_i)$$

or

$$x_{i+1} = rx_i(1 - x_i)$$

[17]Cambel, A B, 'Applied Chaos Theory', Gorman, 2000.

where $r = 1 + k\Delta t$. This is a simple quadratic iterator known as the logistic map and has a range of characteristics depending on the value of r. This is illustrated in Figure 14 which shows the output (for just 30 elements) from this iterator for $r = 1,\ \ r = 2,\ \ r = 3$ and $r = 4$ and for an initial value of 0.1[18]. For $r = 1$ and $r = 2$, convergent behaviour

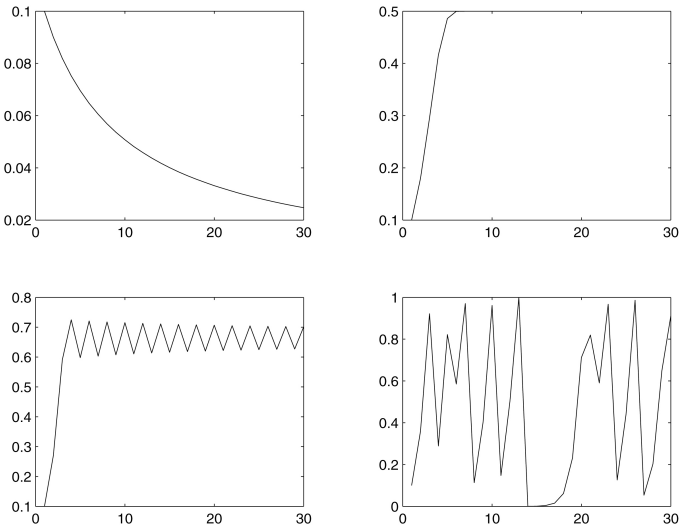


Fig. 14. Output (30 elements) of the logistic map for values of $r = 1$ (top left), $r = 2$ (top right), $r = 3$ (bottom left) and $r = 4$ (bottom right) and an initial value of 0.1.

takes place; for $r = 3$ the output is oscillatory and for $r = 4$ the behaviour is chaotic. The transition from monotonic convergence to oscillatory behaviour is known as a bifurcation and is better illustrated using a so called Fiegenbaum map or diagram which is a plot of the output of the iterator in terms of the values produced (after iterating enough times to produce a consistent output) for different values of r. An example of this for the logistic map is given in Figure 15 for $0 < r \leq 4$ and shows convergent behaviour for values of r from 0 to approximately 3, bifurcations for values of r between approximately 3 and just beyond 3.5 and then a region of chaotic behaviour, achieving 'full chaos' at $r = 4$ where, in each case, the output consists of values between 0 and 1. However, closer inspection of this data representation reveals repeating patterns, an example being given in Figure 16 which is a Fiegenbaum diagram of the output for values of r between 3.840 and 3.855 and values of x between 0.44 and 0.52. As before, we observe a region of convergence, bifurcation and then chaos. Moreover, from Figure 16 we observe another region of this map (for values of r around 3.854) in which this same behaviour occurs. The interesting feature about this map is that the convergence→bifurcation→chaos characteristics are repeated albeit over smaller scales. In other words, there is a similarity of behaviour over smaller scales, i.e. the 'pattern' of behaviour. Further, this complex behaviour comes from a remarkably simple iterator, i.e. the map $x \to rx(1-x)$.

[18]The initial value, which is taken to be any value between 0 and 1, changes the 'signature' of the output but not its characteristics, at least, in the ideal case.

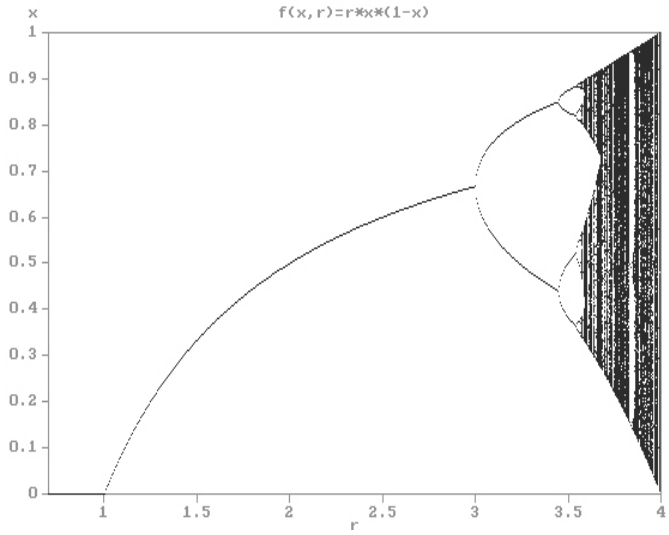


Fig. 15. Feigenbaum diagram of the logistic map for $0 < r < 4$ and $0 < x < 1$.

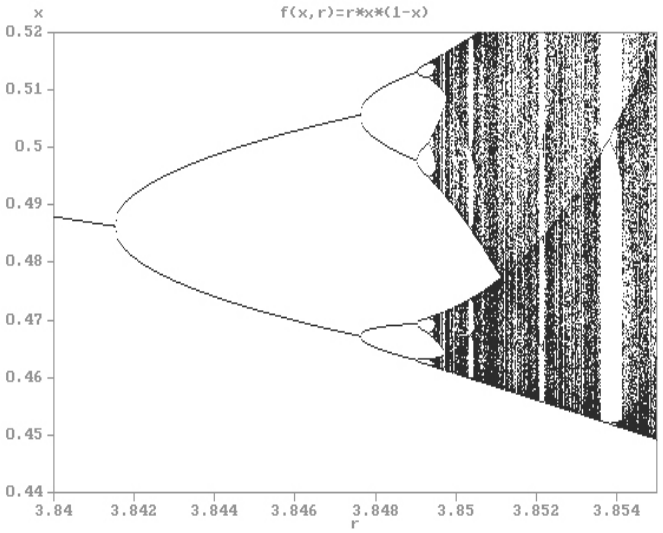


Fig. 16. Feigenbaum diagram of the logistic map for $3.840 < r < 3.855$ and $0.44 < x < 0.52$.

C. Examples of Chaotic Systems

In addition to the logistic map, which has been used in the previous section to introduce a simple IFS that gives a chaotic output, there are a wide variety of other maps which yield signals that exhibit the same basic properties as the logistic map (convergence→bifurcation→chaos) with similar structures at different scales at specific regions of the Feigenbaum diagram. Examples, include the 'maps' given below:

1) Linear functions: The sawtooth map

$$x_{i+1} = 5x_i \mathrm{mod} 4.$$

The tent map

$$x_{i+1} = r(1 - \mid 2x_i - 1 \mid).$$

The generalized tent map

$$x_{i+1} = r(1 - \mid 2x_i - 1 \mid^m), \quad m = 1, 2, 3, ...$$

2) Nonlinear functions: The sin map

$$x_{i+1} = \mid \sin(\pi r x_i \mid .$$

The tangent feedback map

$$x_{i+1} = rx_i[1 - \tan(x_i/2)].$$

The logarithmic feedback map

$$x_{i+1} = rx_i - [1 - \log(1 + x_i)].$$

Further, there are a number of 'variations on a theme' that are of value, an example being the 'delayed logistic map'

$$x_{i+1} = rx_i(1 - x_{i-1})$$

which arises in certain problems in population dynamics. Moreover, coupled iterative maps occur from the development of physical models leading to nonlinear coupled differential equations, a famous and historically important example being the Lorenz equations given by

$$\frac{dx_1}{dt} = a(x_2 - x_1), \frac{dx_2}{dt} = (b - x_3)x_1 - x_2, \frac{dx_3}{dt} = x_1x_2 - cx_3$$

where a, b and c are constants. These equations were originally derived by Lorenz from the fluid equations of motion (the Navier Stokes equation, the equation for thermal conductivity and the continuity equation) used to model heat convection in the atmosphere and were studied in an attempt to explore the transition to turbulence where a fluid layer in a gravitational field is heated from below. By finite differencing these equations, we convert the functions $x_i, \quad i = 1, 2, 3$ into discrete form $x_i^n, \quad i = 1, 2, 3$ giving (using forward differencing)

$$\begin{aligned} x_1^{(n+1)} &= x_1^{(n)} + \Delta t a(x_2^{(n)} - x_1^{(n)}), \\ x_2^{(n+1)} &= x_2^{(n)} + \Delta t[(b - x_3^{(n)})x_1^{(n)} - x_2^{(n)}], \\ x_3^{(n+1)} &= x_3^{(n)} + \Delta t[x_1^{(n)}x_2^{(n)} - cx_3^{(n)}]. \end{aligned}$$

For specific values of a, b and c (e.g. $a = 10, b = 28$ and $c = 8/3$) and a step length Δt, the digital signals $x_1^{(n)}, x_2^{(n)}$ and $x_3^{(n)}$ exhibit chaotic behaviour which can be analysed quantitatively in the three dimension phase space (x_1, x_2, x_3) or variations on this theme, e.g. a three dimensional plot with axes $(x_1 + x_2, x_3, x_1 - x_3)$ or as a two dimensional projection with axes $(x_1 + x_2, x_3)$ an example of which is shown in Figure 17. Here, we see that the path is confined to two domains which are connected. The path is attracted to one domain and then to another but this connection (the point at which the path changes form one domain to the next) occurs in an erratic way - an example of a 'strange attractor'.

As with the simple iterative maps discussed previously, there are a number of nonlinear differential equations (coupled or otherwise) that exhibit chaos the behaviour of which can be quantified using an appropriate phase space. These in include:

The Rössler equations

$$\frac{dx_1}{dt} = -x_2 - x_3, \frac{dx_2}{dt} = x_3 + ax_2, \frac{dx_3}{dt} = b + x_3(x_1 + c).$$

The Hénon-Heiles equations

$$\frac{dx_1}{dt} = p_x, \frac{dp_x}{dt} = -x - 2xy; \frac{dx_2}{dt} = p_y, \quad \frac{dp_y}{dt} = -y - x^2 + y^2.$$

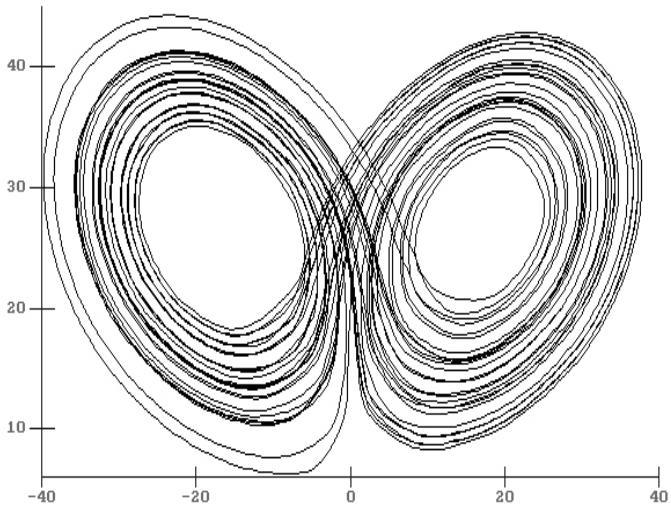


Fig. 17. Two dimensional phase space analysis of the Lorenz equations illustrating the 'strange attractor'.

The Hill's equations

$$\frac{d^2}{dt^2}x(t) + \Omega^2(t)x(t) = 0,$$

a special case being the Mathieu equation when

$$\Omega^2(t) = \omega_0^2(1 + \lambda \cos \omega t),$$

ω_0 and λ being constants. The Duffing Oscillator

$$\frac{dx}{dt} = v, \quad \frac{dv}{dt} = av + x + bx^3 + \cos t$$

where a and b are constants. The non-linear Schrödinger equation

$$\frac{d^2}{dt^2}x(t) + \omega^2 x(t) = \mid x(t) \mid^2 x(t).$$

In each case, the chaotic nature of the output to these systems depends on the values of the constants.

For iterative processes where stable convergent behaviour is expected, an output that is characterised by exponential growth can be taken to be due to unacceptable numerical instability. However, with IFS that exhibit intrinsic instability, in that the output does not converge to a specific value, the Lyapunov exponent is used to quantify the characteristics of the output. This exponent or 'Dimension' provides a principal measure of 'chaoticity' and is derived in Appendix II.

XI. Encryption using Deterministic Chaos

The use of chaos in cryptology was first considered in the early 1950s by the American electrical engineer Claude Shannon and the Russian mathematician Vladimir Alexandrovich Kotelnikov who laid the theoretical foundations for modern information theory and cryptography. It was Shannon who first explicitly mentioned the basic stretch-and-fold mechanism associated with chaos for the purpose of encryption: *Good mixing transformations are often formed by repeated products of*

two simple non-commuting operations [11]. Hopf[19] considered the mixing of dough by such a sequence of non-commuting operations. The dough is first rolled out into a thin slab, then folded over, then rolled, and then folded again and so on. The same principle is used in the making of a Japanese sword, the aim being to produce a material that is a highly diffused version of the original material structure.

The use of chaos in cryptography was not fully appreciated until the late 1980s when the simulation of chaotic dynamical systems became common place and when the role of cryptography in IT became increasingly important. Since the start of the 1990s, an increasing number of publications have considered the use of chaos in cryptography, e.g. [75]-[79]. These have included schemes based on synchronized chaotic (analogue) circuits, for example, which belong to the field of steganography and secure radio communication [80]. Over the 1990s cryptography started to attract a variety of scientists and engineers from diverse fields who started exploiting dynamical systems theory for the purpose of encryption. This included the use of discrete chaotic systems such as the cellular automata, Kolmogorov flows and discrete affine transformations in general to provide more efficient encryption schemes [81]-[84]. Since 2000, the potential of chaos-based communications, especially with regard to spread spectrum modulation, has been recognized. Many authors have described chaotic modulations and suggested a variety of electronics based implementations, e.g. [77]-[80]. However, the emphasis has been on information coding and information hiding and embedding. Much of this published work has been of theoretical and some technological interest with work being undertaken in both an academic and industrial research context (e.g. [85]-[91]). However, it is only relatively recently that the application of chaos-based ciphers have been implemented in software and introduced to the market. One example of this is the basis of the author's own company - Crypstic$^{\mathrm{TM}}$ Limited - in which the principle of multi-algorithmicity using chaos-based ciphers [11], [92] has been used to produce meta-encryption engines that are mounted on a single, a pair or a group of flash (USB - Universal Serial Bus) memory sticks. Some of these memory sticks have been designed to include a hidden memory accessible through a covert procedure (such as the renaming - by deletion - of an existing file or folder) from which the encryption engine(s) can be executed.

Consider an algorithm that outputs a number stream which can be ordered, chaotic or random. In the case of an ordered number stream (those generated from a discretized piecewise continuous functions for example), the complexity of the field is clearly low. Moreover, the information and specifically the information entropy (the lack of information we have about the exact state of the number stream) is low as is the information content that can be conveyed by such a number stream.

A random number stream (taken to have a uniform PDF, for example) will provide a sequence from which, under ideal circumstances, it is not possible to predict any number in the sequence from the previous values. All we can say is that the probability of any number occurring between a specified range is equally likely. In this case, the information entropy is high. However, the complexity of the field, in terms of erratic transitions from one type of localized behaviour to another, is low. Thus, in comparison to a random field, a chaotic field is high in complexity but its information entropy, while naturally higher than an ordered field is lower than that of a random field, e.g. chaotic fields which exhibit uniform number distributions are rare. Such fields therefore need to be post-processed in order that the output conforms to a uniform distribution.

We consider a dynamic continuous-state continuous-time system $\mathcal{S} = \langle X, \mathcal{K}, f \rangle$ as follows [11]:

$$\frac{dx}{dt} = f\left(x, k\right), \quad x \in X, k \in \mathcal{K}$$

where f is a a smooth function, X is a state space and $\mathcal{K}$ is a parameter space. The equation is taken to satisfy the conditions of the existence and uniqueness of solutions $x(x_0, t)$ with the initial condition $x_0 = x\left(x_0, 0\right)$, the solution curve $\varphi_t\left(x_0, t\right)$ being the trajectory.

For cryptography, we focus on dynamic discrete-time systems which can be written in the following form:

$$x_{i+1} = f\left(x_i, k\right), x_i \in X, k \in \mathcal{K}, i = 0, 1, 2, \ldots$$

where x_i is a discrete state of the system. The trajectory $\varphi\left(x_i, x_0\right)$ is defined by the sequence $x_0, x_1, x_2, \ldots$. This equation is similar to the cryptographic iterated functions used for pseudo random number generation, block ciphers and other constructions such as the DES, RSA nd AES ciphers. Consequently, in both nonlinear dynamics and cryptography we deal with an iterated key-dependent transformation of information. There are several sufficient conditions satisfied by a dynamic system to guarantee chaos, the sensitivity to initial conditions and topological transitivity being the most common.

A chaotic continuous-state discrete-time system is a dynamic system $\mathcal{S} = \langle X, f \rangle$ with two properties [92]: (i) given a metric space X and a mapping $f : X \rightarrow X$, then f is topologically transitive on X if, for any two open sets $U, V \subset X$, there is $n \geq 0$ such that $f^n(U) \cap V \neq \emptyset$; (ii) the map f is sensitive to initial conditions if there is $\delta > 0, n \geq 0$ given that for any $x \in X$ and for any neighborhood H_x of x there is $y \in H_x$, such that $|f^n(x) - f^n(y)| > \delta$. These properties can be interpreted as follows: a dynamic system is chaotic if all trajectories are bounded (by the attractor) and nearby trajectories diverge exponentially at every point of the phase space. The trajectories are continuous and belong to a two-dimensional system that is said to be chaotic. This yields to a natural synergy between chaotic and cryptographic systems that can be described in terms of the following: (i) topological transitivity which ensures that the system output covers all the state space, e.g. any plaintext can be encrypted into any ciphertext; (ii) sensitivity to initial condition which corresponds to Shannon's original requirements for an encryption system in the late 1940s. In both chaos and cryptography we are dealing with systems in which a small variation of any variable changes the outputs considerably.

[19]Hopf, Eberhard F. F, (1902-1983), an Austrian mathematician who made significant contributions in topology and ergodic theory and studied the mixing in compact spaces, e.g. *On Causality, Statistics and Probability*, Journal of Mathematics and Physics, 13, 51-102, 1934.

A. Stream Cipher Encryption

The use of discrete chaotic fields for encrypting data can follow the same basic approach as used with regard to the application of pseudo random number generating algorithms for stream ciphers. Pseudo chaotic numbers are in principle, ideal for cryptography because they produce number streams that are ultra-sensitive to the initial value (the key). However, instead of using iterative based maps using modular arithmetic with integer operations, here, we require the application of nonlinear maps using floating point arithmetic. Thus, the first drawback concerning the application of deterministic chaos for encryption concerns the processing speed, i.e. pseudo random number generators typically output integer streams using integer arithmetic whereas pseudo chaotic number generators produce floating point streams using floating point arithmetic. Another drawback of chaos based cryptography is that the cycle length (i.e. the period over which the number stream repeats itself) is relatively short and not easily quantifiable when compared to the cycle length available using conventional PRNGs, e.g. additive generators, which commence by initialising an array x_i with random numbers (not all of which are even) so that we can consider the initial state of the generator to be $x_1, x_2, x_3, ...$ to which we then apply

$$x_i = (x_{i-a} + x_{i-b} + ... + x_{i-m}) \mathrm{mod} 2^n$$

where $a, b, ..., m$ and n are assigned integers[20], have very long cycle lengths of the order of $2^f(2^{55} - 1)$ where $0 \leq f \leq n$ and linear feedback shift registers with the form

$$x_n = (c_1 x_{n-1} + c_2 x_{n-2} + c_m x_{n-m}) \mathrm{mod} 2^k$$

which, for specific values of $c_1, c_2, ...c_m$ have cycle lengths of 2^k.

The application of deterministic chaos to encryption has two distinct disadvantages relative to the application of PRNGs. Another feature of IFS is that the regions over which chaotic behaviour can be generated may be limited. However, this limitation can be overcome by designing IFS with the specific aim of increasing the range of chaos. One method is to use well known maps and modify them to extend the region of chaos. For example, the Matthews cipher is a modification of the logistic map to [93]

$$x_{i+1} = (1 + r)\left(1 + \frac{1}{r}\right)^r x_i(1 - x_i)^r, \quad r \in (0, 4].$$

The effect of this generalization is seen in Figure 18 which shows the Feigenbaum diagram for values of r between 1 and 4. Compared to the conventional logistic map $x_{i+1} = rx_i(1 - x_i), \quad r \in (0, 4]$ which yields full chaos at $r = 4$, the chaotic behaviour of the Matthews map is clearly more extensive providing full chaos for the majority (but not all) of values of r between approximately 0.5 and 4. In the conventional case, the key is the value of x_0 (the initial condition). In addition, because there is a wide range of chaotic behaviour for the Matthews map, the value of r itself can be used as a primary (or secondary) key.

[20]A well known example is the 'Fish generator' $x_i = (x_{i-55} + x_{i-24}) \mathrm{mod} 2^{32}$

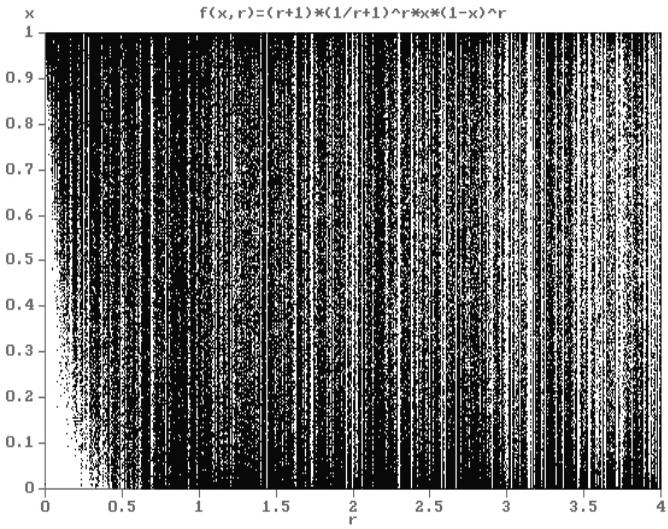


Fig. 18. Feigenbaum map of the Matthews cipher

The approach to using deterministic chaos for encryption has to date, been based on using conventional and other well known chaotic models of the type discussed above with modifications such as the Matthew map as required. However, in cryptography, the physical model from which a chaotic map has been derived is not important; only the fact that the map provides a cipher that is 'good' at scrambling the plaintext in terms of diffusion and confusion. This point leads to an approach which exploits two basic features of chaotic maps: (i) they increase the complexity of the cipher; (ii) there are an unlimited number of maps of the form $x_{i+1} = f(x_i)$, for example, that can be literally 'invented' and then tested for chaoticity to produce a data base of algorithms. However, it is important to stress that such ciphers, once 'invented', need to be post-processed to ensure that the cipher stream is uniformly distributed which, in turn, requires further computational overheads and, as discussed in the following section, may include significant cipher redundancy.

The low cycle lengths associated with chaotic iterators can be overcome by designing block ciphers where the iterator produces a cipher stream only over a block of data whose length is significantly less than that of the cycle length of the iterator, each block being encrypted using a different key and/or algorithm. The use of different algorithms for encrypting different blocks of data provides an approach that is 'multi-algorithmic'.

B. Block Cipher Encryption and Multi-algorithmicity

Instead of using a single algorithm (such as a Matthews cipher) to encrypt data over a series of blocks using different (block) keys, we can use different algorithms, i.e. chaotic maps. Two maps can be used to generate the length of each block and the maps that are used to encrypt the plaintext over each block. Thus, let us suppose we have designed a data base consisting of 100 chaotic maps consisting of IFS $f_1, f_2, f_3, ..., f_{100}$, each of which generates a floating point

number steam through the operation

$$x_{i+1} = f_m(x_i, p_1, p_2, ...)$$

where the parameters $p_1, p_2, ...$ are pre-set or 'hard-wired' to produce chaos for any initial value $x_0 \in (0, 1)$. An 'algorithm selection key' is then introduced in which two algorithms (or possibly the same algorithm) are chosen to 'drive' the block cipher - f_{50} and f_{29} say, the key in this case being $(50, 29)$. Here, we shall consider the case where map f_{50} determines the algorithm selection and map f_{29} determines the block size. Suppose map f_{50} is then initiated with the key 0.26735625, for example, and map f_{29} with the key 0.65376301. The output from these maps (floating point number streams) are then normalized, multiplied by 100 and 1000, respectively, for example, and then rounded to produce integer streams with values ranging from 1 to 100 and 1 to 1000, respectively. Let us suppose that the first few values of these integer streams are $28, 58, 3, 61$ and $202, 38, 785, 426$, respectively. The block encryption starts by using map 28 to encrypt 202 elements of the plaintext using the key 0.78654876 say. The second block of 38 elements is then encrypted using map 58 (the initial value being the last floating point value produced by algorithm 28) and the third block of 785 elements is encrypted using algorithm 3 (the initial value being the last floating point value produced by algorithm 58) and so on. The process continues until the plaintext has been fully encrypted with the 'session key' (50,29,0.26735625,0.65376301,0.78654876).

Encryption is typically undertaken using a binary representation of the plaintext and applying an XOR operation using a binary representation of the cipher stream. This can be constructed using a variety of ways. For example, one could extract the last significant bits from the floating point format of x_i. Another approach, is to divide the floating point range of the cipher into two compact regions and apply a suitable threshold. For example, suppose that the output x_i from a map operating over a given block consists of floating point value between 0 and 1, then, with the application of a threshold of 0.5, we can consider generating the bit stream

$$b(x_i) = \begin{cases} 1, & x_i \in (0.5, 1]; \\ 0, & x_i \in [0, 0.5). \end{cases}$$

However, in applying such a scheme, we are assuming that the distribution of x_i is uniform and this is rarely the case with chaotic maps. Figure 19 shows the PDF for the logistic map $x_{i+1} = 4x_i(1 - x_i)$ which reveals a non-uniform distribution with a bias for floating point numbers approaching 0 and 1. However, the mid range (i.e. for $x_i \in [0.3, 0.7]$) is relatively flat indicating that the probability for the occurrence of different numbers generated by the logistic map in the mid range is the same. In order to apply the threshold partitioning method discussed above in a way that provides an output that is uniformly distributed for a any chaotic map, it is necessary to introduce appropriate conditions and modify the above to the form

$$b(x_i) = \begin{cases} 1, & x_i \in [T, T + \Delta_+); \\ 0, & x_i \in [T - \Delta_-, T); \\ -1, & \text{otherwise.} \end{cases}$$

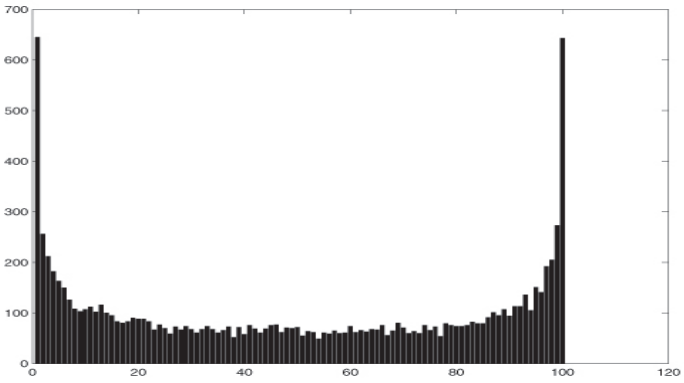


Fig. 19. Probability density function (with 100 bins) of the output from the logistic map for 10000 iterations.

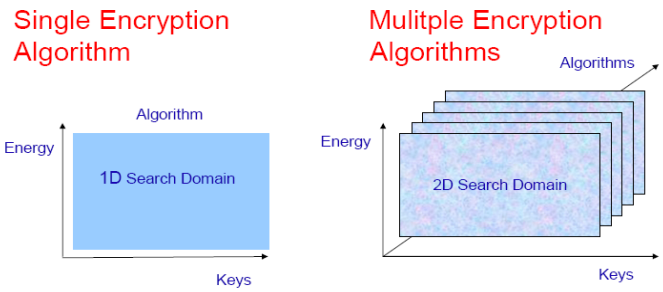


Fig. 20. Illustration of the effect of using multiple algorithms for generating a stream cipher on the computational *Energy* required to attempt a brute force attack.

where T is the threshold and Δ_+ and Δ_- are those values which yield an output stream that characterizes (to a good approximation) a uniform distribution. For example, in the case of the logistic map $T = 0.5$ and $\Delta_+ = \Delta_- = 0.2$. This aspect of the application of deterministic chaos to cryptography, together with the search for a parameter or set of parameters that provides full chaos for an 'invented' map, determines the overall suitability of the function that has been 'invented' for this application.

The 'filtering' of a chaotic field to generate a uniformly distributed output is equivalent to maximizing the entropy of the cipher stream (i.e. generating a cipher stream with a uniform PDF) which is an essential condition in cryptography. In terms of cryptanalysis and attack, the multi-algorithmic approach to designing a block cipher introduces a new 'dimension' to the problem. The conventional problem associated with an attack on a symmetric cipher is to search for the private key(s) given knowledge of the algorithm. Here, the problem is to search not only for the session key(s), but the algorithms they 'drive' as illustrated in Figure 20.

One over-riding issue concerning cryptology in general, is that algorithm secrecy is weak. In other words, a cryptographic system should not rely on the secrecy of its algorithms and all such algorithms should be openly published[21]. The system described here is multi-algorithmic, relying on many different chaotic maps to encrypt the data. Here, publication of the

[21]Except for some algorithms developed by certain government agencies - perhaps they have something to hide!

algorithms can be done in the knowledge that many more maps can be invented as required (subject to appropriate conditions in terms of generating a fully chaotic field with a uniform PDF) by a programmer, or possibly with appropriate 'training' of a digital computer.

XII. CRYPSTIC$^{\mathrm{TM}}$

Crypstic^{*TM} Limited[22] is the trade mark for a USB based product that currently uses three approaches for providing secure mobile information exchange: (i) obfuscation; (ii) disinformation; (iii) multi-algorithmic encryption using chaos. The product is currently being marketed through Titon International Limited (http://www.titoninternational.co.uk/). In addition to performing basic encryption/decryption of data, Crypstic$^{\mathrm{TM}}$ is also used to covertly transfer digital images or other plaintext that has been converted into a digital image via the technique discussed in Section VII. In this case, the noise field (see Figure 5) is the cipher output by Crypstic$^{\mathrm{TM}}$.

A. Obfuscation and Disinformation

Obfuscation is undertaken by embedding the application (the *.exe* file that performs the encryption/decryption) in an environment (i.e. the USB memory) that contains a wealth of data (files and folders etc.) that is ideally designed to reflect the users portfolio. This can includes areas that are password protected and other public domain encryption systems with encrypted files, as required, that may be broken and even generate apparently valuable information (given a successful attack) but are, in fact, provided purely as a form of disinformation. This environment is designed in order to provide a potential attacker, who has gained access to a users Crypstic$^{\mathrm{TM}}$ through theft, for example, with a 'target rich' environment. The rationale associated with the use of a Crypstic$^{\mathrm{TM}}$ as a mobile encryption/decryption device follows that associated with a user's management of a 'key ring'. In other words, it is assumed that the user will maintain and implement the Crypstic$^{\mathrm{TM}}$ in the same way as a conventional set of keys are used. However, in the case of loss or theft, a new Crypstic$^{\mathrm{TM}}$ must be issued which includes a new encryption engine and under no circumstances is the original Crypstic$^{\mathrm{TM}}$ re-issued. Management of the encryption engines and their distribution is, of course, undertaken by Crypstic$^{\mathrm{TM}}$ Limited which maintains a data base of current users and the encryption engines provided to them in compliance with the RIP Act, 2000, Section 49, which deals with the power of disclosure, i.e. for Crypstic$^{\mathrm{TM}}$ Limited to provide the appropriate encryption engine for the decryption of any encrypted data that is under investigation by an appropriate authority.

B. Encryption Engine Design

The encryption engine itself can be based on any number of algorithms, each algorithm having been 'designed' with respect to the required (maximum entropy) performance conditions through implementation of the appropriate threshold parameters T and $\Delta_{\pm}$. The design is based on applying the following basic steps:

Step 1: Invent a (non-linear) function f and apply the IFS $x_{i+1} = f(x_i, p_1, p_2, ...)$

Step 2: Normalise the output of the IFS so that $\mathbf{x}_{\infty} = 1$.

Step 3: Graph the output x_i and adjust parameters $p_1, p_2, ...$ until the output 'looks' chaotic.

Step 4: Graph the histogram of the output and observe if there is a significant region of the histogram over which it is 'flat'.

Step 5: Set the values of the thresholds T and $\Delta_{\pm}$ based on 'observations' made in Step 4.

Analysing of the ISF using a Feigenbaum diagram can also be undertaken but this can be computationally intensive. Further, each ISF can be categorised in terms of parameters such as the Lyapunov Dimension (Appendix II) and information entropy, for example. However, in practice, such parameters yield little in terms of the design of an IFS and are primarily 'academic'. Indeed, the invention and design of such algorithms has a certain 'art' to it which improves with experience. It should be noted that many such inventions fail to be of value in that the statistics may not be suitable (e.g. the PDF may not be flat enough or is flat only over a very limited portion of the PDF), chaoticity may not be guaranteed for all values of the seed x_0 between 0 and 1 and the numerical performance of the algorithm may be poor. The aim is to obtain a simple IFS that is numerically relatively trivial to compute, has a broad statistical distribution and is valid for all floating point values of x_0 between 0 and 1. Examples of the IFS used for Crypstic$^{\mathrm{TM}}$ are given in the following table where the values of T, Δ_+ and Δ_- apply to the normalised output stream generated by each function.

Function $f(x)$	r	T	Δ_+	Δ_-
$rx(1-\tan(x/2))$	3.3725	0.5	0.3	0.3
$rx[1-x(1+x^2)]$	3.17	0.5	0.25	0.35
$rx[1-x\log(1+x)]$	2.816	0.6	0.3	0.2
$r(1-\mid 2x-1\mid^{1.456})$	0.9999	0.5	0.3	0.3
$\mid \sin(\pi r x^{1.09778}) \mid$	0.9990	0.6	0.25	0.25

The functions given in the table above produce outputs that have a relatively broad and smooth histogram which can be made flat by application of the values of T and $\Delta_{\pm}$. Some functions, however, produce poor characteristic in this respect. For example, the function

$$f(x) = r \mid 1 - \tan(\sin x) \mid, \quad r = 1.5$$

has a highly irregular histogram (see Figure 21) which is not suitable in terms of applying values of T and $\Delta_{\pm}$ and, as such, is not an appropriate IFS for a Crypstic$^{\mathrm{TM}}$ application.

C. Graphical User Interface

In conventional encryption systems, it is typical to provide a Graphical User Interface (GUI) with fields for inputting the plaintext and outputting the ciphertext where the name

[22]Incorporated under the Companies Act 1985 for England and Wales as a Private Company, Limited on 19th January, 2005, Company Number: 5337521

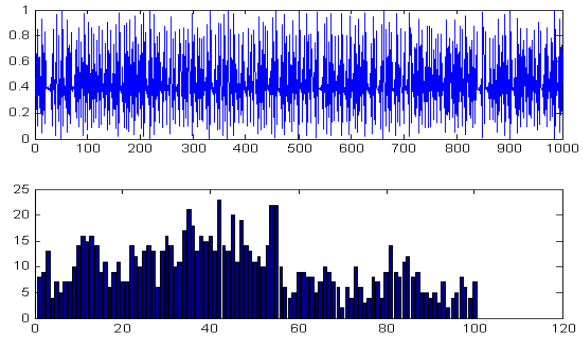


Fig. 21. The first 1000 elements for $x_{i+1} = r \mid 1 - \tan(\sin x_i) \mid, r = 1.5, 0 < x_0 < 1$ (above) and associated histogram (below).

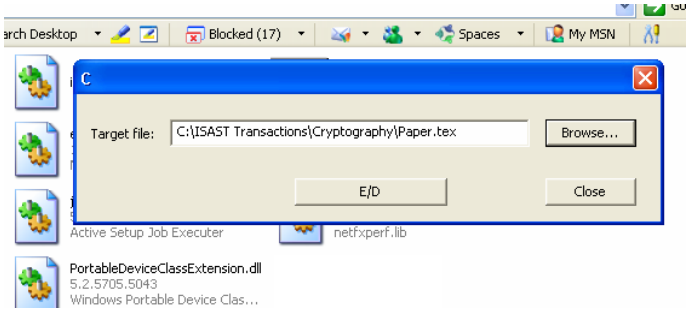


Fig. 22. GUI of Crypstic$^{\mathrm{TM}}$ encryption application.

of the output (including file extension) is supplied by the user. Crypstic$^{\mathrm{TM}}$ outputs the ciphertext by overwriting the input file. This allows the file name, including the extension, to be used to 'seed' the encryption engine and thus requires that the name of the file remains unchanged in order to decrypt. The seed is used to initiate the session key discussed in Section XI(B). The file name is converted to an ASCII 7-bit decimal integer stream which is then concatenated and the resulting decimal integer used to seed a hash function whose output is of the form (d, d, f, f, f) where d is a decimal integer and f is a 32-bit precision floating point number between 0 and 1.

The executable file is camouflaged as a *.dll* file which is embedded in a folder containing many such *.dll* files. The reason for this is that the structure a *.dll* file is close to that of a *.exe* file. Nevertheless, this requires that the source code must be written in such a way that all references to its application are void as discussed in Section II(E). This includes all references to the nature of the data processing involved including words such as *Encrypt* and *Decrypt*, for example[23], so that the compiled file, although camouflaged as a *.dll* file, is forensically inert to attacks undertaken with systems such a WinHEX [94]. In other words, the source code should be written in a way that is 'incomprehensible', a condition that is consistent with the skills of many software engineers! This must include the development of a run time help facility. Clearly, such criteria are at odds with the 'conventional wisdom' associated with the development of applications but the purpose of this approach is to develop a forensically inert executable file that is obfuscated by the environment in which it is placed. An example of the GUI is given in Figure 22.

D. Procedure

The approach to loading the application to encrypt/decrypt a file is based on renaming the *.dll* file to an *.exe* file with a given name as well as the correct extension. Simply renaming a *.dll* file in this way can lead to a possible breach of security by a potential attacker using a key logging system [95]. In order to avoid such an attack, Crypstic$^{\mathrm{TM}}$ uses an approach in which the name of the *.dll* file can be renamed to a *.exe* file by using a 'deletion dominant' procedure. For example, suppose the application is called *enigma.exe*, then by generating a *.dll* file called *engine_gmax_index.dll*, renaming can be accomplished by deleting (in the order given) *lld.* followed by *dni_x* followed by *_en* followed by *g* and then inserting a . between *ae* and including *e* after *ex*. A further application is required such that upon closing the application, the *.exe* file is renamed back to its original *.dll* form. This includes ensuring that the time and date stamps associated with the file are not updated.

The procedure described above is an attampt to obfuscate the use of passwords which are increasingly open to attack [18] especially with regard to password protected USB memory sticks. Many manufacturers break all the rules when attempting to implement security. Checking the password and unlocking the stick are two separate processes, both initiated from the PC. Thus, from the point of view of the stick, they are both separate processes, but this is a major flaw. The best sticks handle all the encryption to and from the flash themselves and do not keep a password at all. The fact that the data cannot be decrypted without it makes it safe. The mediocre sticks store a password inside the flash-controller and check it against a password sent by the PC before unlocking the flash-memory. This way, the password cannot be found by reading out the flash-chip manually. The bad sticks do the same but store the password on flash. Some sticks are even worse than this, they store the password on flash and let the PC do the validation.

In addition to the procedures associated with password validation, the concept of passwords protection is becoming increasingly redundant. For example, Elcomsoft Limited recently filed a US patent for a password cracking technique that relies on the parallel processing capabilities of modern graphics processors. The technique increases the speed of password cracking by a factor of 25 using a GeForce 8800 Ultra graphics card from Nvidia. '*Cracking times can be reduced from days or hours to minutes in some instances and there are plans to introduce the technique into password cracking products*' (http://techreport.com/discussions.x/13460).

E. Protocol

Crypstic$^{\mathrm{TM}}$ is a symmetric encryption system that relies on the user working with a USB memory stick and maintaining a protocol that is consistent with the use of a conventional set of keys, typically located on a key ring. The simplest use of Crystic$^{\mathrm{TM}}$ is for a single user to be issued with a Crypstic$^{\mathrm{TM}}$ which incorporates an encryption engine that is unique (through the utilisation of a unique set of algorithms

[23]Words that can be replaced by E and D respectively in a GUI.

which is registered with Crypstic^{TM} Limited for a given user). The user can then use the Crypstic^{TM} to encrypt/decrypt files and/or folders (after application of a compression algorithm such as *pkzip*, for example) on a PC before closure of a session. In this way, the user maintains a secure environment using a unique encryption engine with a 'key' that includes a covert access route, coupled with appropriate disinformation as discussed in previous sections. Different encryption engines can be incorporated that are used to encrypt disinformation in order to provide a solution to the 'gun to the head problem' as required.

In the case of communications between 'Alice and Bob', both users are issued with crypstics that have encryption engines unique to Alice and Bob, each of whom can use the facility for personal data security as above, and, in addition, can encrypt files for email communications. If any crypstic, by any party, is lost, then a new pair of crypstics are issued with new encryption engines unique to both parties. In addition to a two-party user system, crypstics can be issued to groups of users in a way that provides an appropriate access hierarchy as required.

F. Application of Crypsic^{TM} for Data Protection

There are two specific applications of Crypstic^{TM}. The first is for transmitting encrypted information via the Internet (as an email attachment) using non-standard ciphers and the second is for encrypting information contained in a database maintained on a PC, for example. Database encryption is particularly important in light of the encreasing number of breaches of security concerning sensitive personal information, stored on databases held by organisations through loss/theft of a PC, CD and other storage media. For example, on January 22, 2008, Tom Newton Dunn, the Defence Editor for *The Sun* (UK) published the following: 'A total of three MoD laptops packed full of personal data are now missing, Defence Secretary Des Browne admitted yesterday. He revealed that two laptops had been stolen from blundering officers before a third was taken in Birmingham earlier this month. And none of the sensitive data was encrypted leaving British troops under threat. Mr Brownes announcement to MPs followed last weeks news that a laptop had been stolen from a junior Navy officers car in Birmingham on January 9, 2008. The Tories said an astonishing 347 MoD laptops had been stolen in the past four years. And it emerged that the MoD ignored key recommendations from a Government review to secure all personal data after the scandal of lost child benefit details last year. Last night Whitehall officials were banned from taking laptops containing sensitive information out of their offices. The two previous MoD thefts were in Edinburgh in 2005 and Manchester in 2006. The Manchester computer had the same Navy and RAF database that was nicked in Birmingham details of 600,000 potential recruits in the last ten years. Mr Browne said it included passport, National Insurance, NHS and driving licence numbers, and some banking details. Ordering a full review, he said: 'This must never happen again.' The officers in charge of the laptops may be prosecuted. Tory defence chief Liam Fox slammed MoD 'incompetence'.

Databases and the data they contain remain tempting targets for hackers and internal users, who look to exploit the many widespread weaknesses found in database-driven applications. The following five database-related vulnerabilities are among the most common: (i) password policies; (ii) SQL injection; (iii) cross-site scripting; (iv) data leakage; (v) improper error handling. In a study released in October 2006, the Ponemon Institute [96] found that data breaches cost companies an average of $182 per compromised record, a 31% increase over 2005. Ponemon studied 31 companies that experienced a data breach. The total costs for each loss ranged from less than $1 million to more than $22 million, according to the 2006 findings. 'The difference in security budgets between companies that have been breached or not breached is big. A company that hasn't suffered a breach might have a budget of $500,000 dollars. A company that has suffered a breach will more likely have a budget of $5 million'.

1) Passwords: Passwords, even if properly implemented, are a growing vulnerability as a result of keylogging. A recent White Paper from McAfee states: '... there has been a massive increase in the use of keyloggers, malicious programs that track the user's typing activity to capture passwords and other private information. Between January 2004 and May 2006, keylogger use rose 250% . Meanwhile, the number of fishing alerts tracked by the Anti-fishing Working Group increased 100-fold during the same period.' The paper also added, '...identity theft is taking a high toll on economies around the world', and pointed to a Federal Trade Commission assessment that the annual cost for consumers and businesses in the United States alone is $ 50 billion a year. 'In the United Kingdom, the Home Office has calculated the cost of identity theft to the British economy at $3.2 billion during the last three years, and some estimates from the Australian Centre for Policing Research place the cost of identity theft at $3 billion each year.' According to Bill Gates, 'Passwords are the weak link'. 'We need to move in the direction of smart cards, and multi-factor authentication must be built into the system itself. We need the ability to track what goes on and have a built-in recovery system.'

2) Internal Theft: A fundamental principal concerning internal theft is that you cannot trust your own employees. A recent survey of 1,000 workers by Tickbox.net showed 60% admitted to stealing from their employers confidential documents, sales leads, customer databases and business contacts. A recent Data Genetics International report said that of the cases they had investigated around 70% of data breaches had been internal. A new threat is Pod slurping. This is a method that illegally uploads gigabytes of confidential information from an organization's computer systems to an iPod or any other removable storage device. Those engaging in the practice often utilize programs like *slurp.exe*, which make it easier to search relevant directories on a computer system for typical business documents in Word and Excel format. The *slurp.exe* program is capable of copying 100 MB worth of data from the Windows 'Documents and Settings' directory in a matter of minutes. Once the information is slurped it can be downloaded, analyzed and even sold.

3) Hackers: Hacking is a continual threat. For example, in December 2006, a hacker gained access to a computer system at the University of California, Los Angeles. About 800,000 potential victims were notified. In August, AT& T notified about 19,000 customers that their personal data was compromised after digital miscreants hacked one of its computer systems and gained access to credit card information and other personal data. In late 2005, a timeshare unit of Marriott International Inc. notified over 200,000 customers that data on backup tapes were stolen. In January 2007, retailer TJX Companies Inc., which runs several discount clothing and home goods stores, said that its systems had been breached by an attacker who may have stolen the credit card data of millions of customers.

4) Principal Issues for Solutions Development: Some of the principal issues associated with loss of sensitive data are as follows: (i) Any live database cannot be encrypted otherwise it cannot be queried. Therefore, the first step is to ensure that both physical and electronic security for the database is of the highest level. Any backup tapes must be very securely encrypted. (ii) It is not sensible or acceptable, bearing in mind the internal threats, simply to rely on Intranet or Internet protocols to protect sensitive data. (iii) It is no longer sensible to rely on passwords. (iv) Access to information based on data is essential for a business to operate. However, the risk of data loss seems to be even greater from within the organisation than from external threats. Therefore, there is no sense in securing the perimeter - the same strong access controls need to be in place both internally and externally. This has the added advantage of allowing authorised access externally using the same methodology. The GCHQ project on Information Assurance is moving away from the PKI approach to identity assurance which does not rely on a relatively weak asymmetric encryption system. There seems to be a need to rethink the approach to solving this problem.

5) Database Security and Access: In light of the above, we consider the following points with regard to the application of Crypstic$^{\mathrm{TM}}$ for database security and access: (i) A live database needs to be both electronically and physically secure. (ii) New data needs to be added in such a way that the originating system deletes it after confirming it has been stored safely. (iii) Backup tapes must be both strongly encrypted and safely stored. (iv) Information gathered from the Database by Query Engines must be carefully restricted and each query must leave an audit trail and be capable of being monitored in real time. (v) It must be impossible to ask a Query Engine to recreate all or part of the database - only higher level information analysis or appropriate parts of individual records should be accessible. (vi) Non-password access is essential to defeat keyloggers. Crypstic$^{\mathrm{TM}}$ has developed two new approaches to achieving this. The stronger is based on reconfiguring the hardware within a USB Stick Device. (vii) To identify a user, a biometric would be ideal. A fingerprint reader on a USB Stick would be fine provided a hacker cannot simply emulate the 'match OK' signal. An alternative may be face recognition. For example, Aladdin has a time-sensitive security system. Instead of displaying a code, this could be internally used to create time synchronised encryption. (viii) To defeat a sustained attack on the encryption system, Crypstic$^{\mathrm{TM}}$ provides multi-algorithmic technology whereby every user or group of users has a unique encryption engine. (ix) Loss or theft of the USB Device must not automatically lead to a security breach. However, a security recovery system should be implemented to replace all group devices and internal encryption promptly.

XIII. Discussion

The material discussed in this paper has covered some of the basic principles associated with cryptography in general, including the role of diffusion and confusion for designing ciphers that have no statistical bias. This has been used as a guide in the design of ciphers that are based on the application of IFS exhibiting chaotic behaviour. The use of IFS allow for the design of encryption engines that are multi-algorithmic, each algorithm being based on an IFS that is invented, subject to the condition that the output stream has a uniform PDF. The principle of multi-algorithmicity has been used to develop a new product - Crypstic$^{\mathrm{TM}}$ - that is based on the following: (i) a multi-algorithmic block encryption engine consisting of a unique set of IFS; (ii) maximum entropy conversion to a bit stream cipher; (iii) a key that is determined by the file name to be encrypted/decrypted. The approach has passed all statistical tests [11] recommended by the National Institute of Standards and Technology (NIST) [97].

Access and use of the encryption engine is based on utilizing a commercial-off-the-shelf USB flash memory via a combination of camouflage, obfuscation and disinformation in order to elude any potential attacker. The approach has been based on respecting the following issues: (i) security is a process not a product; (ii) never underestimate the enemy; (iii) the longer that any cryptosystem, or part thereof, remains of the same type with the same function, the more vulnerable the system becomes to a successful attack. Point (iii) is a singularly important feature which Crypstic$^{\mathrm{TM}}$ overcomes by utilizing a dynamic approach to the design and distribution of encryption engines.

A. Chaos Theory and Cryptography

There is a fundamental relationship between cryptography and chaos. In both cases, the object of study is a dynamic system that performs an iterative nonlinear transformation of information in an apparently unpredictable but deterministic manner [11]. In terms of sensitivity to initial conditions together with the mixing properties of chaotic systems, with appropriate entropy conscious post-processing (as discussed in Section VIII), it is possible to ensure cryptographic confusion and diffusion as discussed in Section V. However, there are a number of conceptual differences between chaos theory and cryptography: (i) chaos theory is often concerned with the study of dynamical systems defined on an infinite state space whereas cryptography relies on a finite-state machine and all chaos models implemented on a computer are approximations, i. e. digital computers can only generate pseudo-chaos; (ii) chaos theory typically studies the asymptotic behaviour of a nonlinear system (i.e. the behaviour of the system as the number of iterations approach infinity when the Lyapunov

dimension can be quantified - see Appendix II in which the definition of the Lyapunov dimension is based on $N \rightarrow \infty$), whereas cryptography focuses on the effect of a small number of iterations that are typically determined by the size of the plaintext; (iii) chaos theory is not necessarily concerned with the algorithmic complexity of an IFS but in the interpretation of the IFS with regard to the physical model from which it has been derived; in cryptography, complexity is the key issue and thus, the concepts of cryptographic security and efficiency have no counterparts in chaos theory; (iv) classical chaotic systems usually have recognizable attractors whereas in cryptography, we attempt to eliminate any structure by post processing the output to produce a maximum entropy cipher; (v) unlike chaos in general, cryptographic systems use a combination of independent variables to provide an output that is unpredictable to an observer; (vi) chaos theory is often associated with the mathematical model used to quantify a physically significant problem, whereas in cryptography, the physical model is of no importance. The following table provides a comparison between chaos theory and cryptography in terms of those aspects of the two subjects that have been considered in this paper [11].

Chaos Theory	**Cryptography**
Chaotic system	Pseudo-chaotic system
Nonlinear transform	Nonlinear transform
Infinite number of states	Finite states
Infinite number of iterations	Finite iterations
Initial state	Plaintext
Final state	Ciphertext
Initial condition(s) and/or parameter(s)	key
Asymptotic independence of initial and final states	Confusion
Sensitivity to initial condition(s) and parameter(s) mixing	Diffusion

Point (vi) above is of particular importance with regard to the design of chaos based encryption engines. Whereas previous publications in this field (e.g. [75], [76], [82], [83], [93]) have considered variations on established IFS, in this paper, we have considered the idea that, in principal, an unlimited number of IFS can be 'invented' by a designer in order to provide a limitless range of multi-algorithmic encryption engines.

Chaotic systems are algorithmically random and thus, cannot be predicted by a deterministic machine even with infinite power [11], [92]. However, chaotic systems are predictable by a probabilistic machine and thus, finding probabilistically unpredictable chaotic systems is a central problem for chaos based cryptography. In this paper, the generation of an unpredictable cipher has been undertaken by filtering those numbers that belong to a uniform partition of the PDF in order to generate a maximum entropy cipher. This approach comes at the expense of numerical performance since a relatively large percentage (upto 50% is some cases) of the floating point numbers that are computed are consequently discarded by the filter.

Chaos theory is not related to number theory in the same way as conventional cryptographic algorithms are, nor is chaos theory related to the computational complexity analysis that underpins digital cryptography. Hence, neither chaos, nor pseudo-chaos can guarantee pseudo-randomness and resistance to different kinds of cryptanalysis based on conventional scenarios. The use of floating-point arithmetic is the most obvious solution for approximating continuous chaos on a finite-state machine. However, there is no straightforward application to pseudo-random number generation and cipher generation. Critical problems can include: (i) growing rounding-off errors; (ii) structural instability, i.e. different initial conditions and parameters giving different cryptographic properties, such as very short cycles, weak plaintexts or weak keys.

Chaotic systems based on smooth nonlinear functions (e.g. x^2, $\sin(x)$, $\tan(x)$ and $\log(x)$) produce sequences with a highly non-uniform distribution and can therefore be predicted by a probabilistic machine. By applying a partitioning strategy to generate a uniform output, a bit stream cipher with uniform statistical properties can be obtained which passes all pseudo-randomness tests. Some piecewise-linear maps generate sequences, which have theoretically flat distributions. However, in practice, these maps are less suitable than nonlinear maps because of the overall effect of linearity, rounding and iterative transformations and may be characterised by highly non-uniform statistics. The need to post-process the outputs form chaotic iterators, in order to provide bit-streams with no statistical bias, leads to a cryptosystem that is relatively inefficient when compared to conventional PRNGs. Further, the lack of any fundamental theoretical framework with regard to the pseudo-random properties of IFS leads to a basic incompatibility with modern cryptography. However, this is off-set by the flexibility associated with the use of multi-algorithmicity for generating numerous and, theoretically, an unlimited number of unique encryption engines.

All conventional cryptographic systems (encryption schemes, pseudo-random generators, hash functions) can be considered to be binary pseudo-chaotic systems, based on bit stream encryption defined over a finite space. Such systems are periodic and have a limited sensitivity to the initial conditions, i.e. the Lyapunov exponents are positive only if measured at the beginning of the process (before one can see the cycles). Different IFS will have different cycle lengths. Measuring the minimum, maximum and average length of an IFS is not a trivial problem, but clearly, ideal cryptographic systems have a single cycle that includes all the possible states. In practice, the cycle length of a IFS must be less than the length of the plaintext for which it is used. In this sense, it is arguable that the use of multi-algorithmicity is a necessary as well as a desirable design feature.

Iterative block ciphers can be viewed in terms of a combination of two linked pseudo-chaotic systems, data and round-key systems. The IFS of such systems includes nonlinear substitutions, row shifts, column mixing and so on. The round-key system is a pseudo-random generator providing a sensitivity dependence of the ciphertext on the key. Technically, most

pseudo-random generators are based on the stretch-and-fold transformation: first, the state is stretched over a large space (e.g. through multiplication and/or through application of a power), then folded into the original state space (using a periodic function, for example). This stretch-and-fold transformation forms the basis of the majority of IFS used in cryptography.

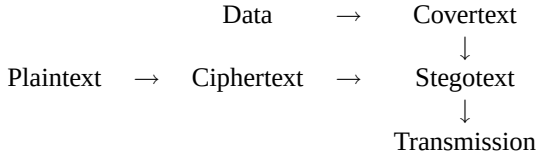
In the design of any chaos based cryptosystem, it is important to have a structurally stable cryptosystem, i.e. a system that has (almost) the same cycle length and Lyapunov exponents for all initial conditions and a given control parameter set. Many pseudo-chaotic systems do not possess this quality. Approximations to chaos are usually based on fixed precision computations. However, it is possible to increase the precision or resolution (e.g. the length of a binary state string) in each iteration, a precision that can, according to a set of rules, be used to estimate the impact of an error. One-way transformations form the basis of most PRNGs, whereas a key-dependent invertible transformation is the essence of a cipher or encryption scheme. Most chaos based ciphers can be extended to include invertible transformations such as XOR, cyclic shifts and other permutations and the latter transformations can also be considered as pseudo-chaotic maps. Further, asymmetric cryptographic systems are based on trapdoor functions, i.e. functions that have a one-way property unless a secret parameter (trapdoor) is known. No counterpart of a trapdoor transformation is known in chaos theory and thus, it is not currently possible to produce an equivalent to the RSA algorithm, for example, using an IFS. However, it is noted that asymmetric encryption algorithms such as the RSA algorithm can be used to transfer a database of algorithms used for the multi-algorithmic symmetric encryption schemes considered in this paper.

B. Covertext and Stegotext

One of the principal weaknesses of all encryption systems is that the form of the output data (the ciphertext), if intercepted, alerts the intruder to the fact that the information being transmitted may have some importance and that it is therefore worth attacking and attempting to decrypt it. In Figure 1, for example, if a postal worker observed some sophisticated 'strong box' with an impressive lock passing through the post office, it would be natural for them to wonder what might be inside. It would also be natural to assume that the contents of the box would have a value in proportion with the strength of the box/lock. These aspects of ciphertext transmission can be used to propagate disinformation, achieved by encrypting information that is specifically designed to be intercepted and decrypted. In this case, we assume that the intercept will be attacked, decrypted and the information retrieved. The key to this approach is to make sure that the ciphertext is relatively strong (but not too strong!) and that the information extracted is of high quality in terms of providing the attacker with 'intelligence' that is perceived to be valuable and compatible with their expectations, i.e. information that reflects the concerns/interests of the individual(s) and/or organisation(s) that encrypted the data. This approach provides the interceptor with a 'honey pot' designed to maximize their confidence especially when they have had to put a significant amount of work in to 'extracting it'. The trick is to make sure that this process is not too hard or too easy. 'Too hard' will defeat the object of the exercise as the attacker might give up; 'too easy', and the attacker will suspect a set-up!

In addition to providing an attacker with a honey-pot for the dissemination of disinformation, it is of significant value if a method can be found that allows the real information to be transmitted by embedding it in non-sensitive information after (or otherwise) it has been encrypted, e.g. camouflaging the ciphertext using methods of *Steganography*. This provides a significant advantage over cryptography alone in that encrypted messages do not attract attention to themselves. No matter how well plaintext is encrypted (i.e. how unbreakable it is), by default, a ciphertext will arouse suspicion and may in itself be incriminating, as in some countries encryption is illegal. With reference to Figure 1, Steganography is equivalent to transforming the 'strong box' into some other object that will pass through without being noticed - a 'chocolate-box', for example.

The word 'Steganography' is of Greek origin and means 'covered', or 'hidden writing'. In general, a steganographic message appears as something else or *Covertext*. The conversion of a ciphertext to another plaintext form is called *Stegotext* conversion and is based on the use of covertext. Some covertext must first be invented and the ciphertext embedded or mapped on to it in some way to produce the stegotext. The basic principle is given in the following schematic diagram:



Note that this approach does not necessarily require the use of plaintext to ciphertext conversion as illustrated above and that plaintext can be converted into stegotext directly. For example, a simple approach to this is to use a mask to delete all characters in a message except those that are to be read by the recipient of the message. Apart from establishing a method of exchanging the mask, which is equivalent to the key in cryptography, the principal problem with this approach is that different messages have to be continuously 'invented' in order to accommodate hidden messages and that these 'inventions' must appear to be legitimate. However, the wealth of data that is generated and transmitted in todays environment, and the wide variety of formats that are used, means that there is much greater potential for exploiting steganographic methods than were previously available. In other words, the wealth of information now available has generated a camouflage rich environment in which to operate and one can attempt to hide plaintext or ciphertext (or both) in a host of data types, including audio and video files and digital images. Moreover, by understanding the characteristics of a transmission environment, it is possible to conceive techniques in which information can be embedded in the transmission noise, i.e. where natural transmission noise is the covertext. There are,

of course, a range of counter measures - *Steganalysis* - that can be implemented in order to detect stegotext. However, the techniques usually require access to the covertext which is then compared with the stegotext to see if any modifications have been introduced. The problem is to find ways of obtaining the original stegotext which is equivalent to a plaintext attack.

C. Hiding Data in Images

The relatively large amount of data contained in digital images makes them a good medium for undertaking steganography. Consequently digital images can be used to hide messages in other images. A colour image typically has 8 bits to represent the red, green and blue components. Each colour component is composed of 256 colour values and the modification of some of these values in order to hide other data is undetectable by the human eye. This modification is often undertaken by changing the least significant bit in the binary representation of a colour or grey level value (for grey level digital images). For example, the grey level value 128 has the binary representation 10000000. If we change the least significant bit to give 10000001 (which corresponds to a grey level value of 129) then the difference in the output image, in terms of a single pixel, will not be discernable. Hence, the least significant bit can be used to encode information through modification of pixel intensity. Further, if this is done for each colour component, then a letter of ASCII text can be represented for every three pixels. The larger the host image compared with the hidden 'image', the more difficult it is to detect the message. Further, it is possible to hide an image in another image for which there are a number of approaches available.

Crypstic$^{\mathrm{TM}}$ explicitly uses the method discussed in Section VII on Stochastic Diffusion for steganographic applications. The plaintext (which, in the case of written material, is limited in this application to an image of a single text page) is first converted into an image file which is then diffused with a noise field that is generated by Crypstic$^{\mathrm{TM}}$. The host image (which is embedded in an environment of different digital images) is distributed with each Crypstic$^{\mathrm{TM}}$ depending on the protocol and user network associated with its application. Note that the host image represents, quite literally, the key to recovering the hidden image. The additive process applied is equivalent to the process of confusion that is the basis for a substitution cipher. Rather than the key being used to generate a random number stream using a pre-defined algorithm from which the stream can be re-generated (for the same key), the digital image is, in effect, being used as the cipher. By diffusing the image with a noise field, it is possible to hide the output in a host image without having to resort to quantization. In the case of large plaintext documents, each page is converted into an image file and the image stream embedded in a host video.

D. Hiding Data in Noise

The 'art' of steganography is to use what ever covertext is readily available to make the detection of plaintext or, ideally, the ciphertext as difficult as possible. This means that the embedding method used to introduce the plaintext/ciphertext into the covertext should produce a stegotext that is indistinguishable from the covertext in terms of its statistical characteristics and/or the information it conveys. From an information theoretic point of view, the covertext should have significantly more capacity than the cipheretext, i.e. there must be a high level of redundancy. Utilising noisy environments often provides an effective solution to this problem. There are three approaches that can be considered: (i) embedding the ciphertext in real noise; (ii) transforming the ciphertext into noise that is then added to data; (iii) replacing real noise with ciphertext that has been transformed in to synthetic noise with exactly the same properties as the real noise.

In the first case, we can make use of noise sources such as thermal noise, flicker noise, and shot noise associated with electronics that digitize an analogue signal. In digital imaging this may be noise from the imaging Charge Couple Device (CCD) element; for digital audio, it may be noise associated with the recording techniques used or amplification equipment. Natural noise generated in electronic equipment usually provides enough variation in the captured digital information so that it can be exploited as a noise source to 'cover' hidden data. Because such noise is usually a linear combination of different noise types generated by different physical mechanisms, it is usually characterised by a normal or Gaussian distribution as a result of the Central Limit Theorem (see Appendix I).

In the second case, the ciphertext is transformed into noise whose properties are consistent with the noise that is to be expected in certain data fields. For example, lossy compression schemes (such as JPEG - Joint Photographic Expert Group) always introduce some error (numerical error) into the decompressed data and this can be exploited for steganographic purposes. By taking a clean image and adding ciphertext noise to it, information can be transmitted covertly providing all users of the image assume that it is the output of a JPEG or some other lossy compressor. Of course, if such an image is JPEG compressed, then the covert information may be badly corrupted.

In the third case, we are required to analyse real noise and derive an algorithm for its synthesis. Here, the noise has to be carefully synthesized because it may be readily observable as it represents the data stream in its entirety rather than data that is 'cloaked' in natural noise. This technique also requires that the reconstruction/decryption method is robust in the presence of real noise which we should assume will be added to the synthesized noise during a transmission phase. In this case, random fractal models are of value because the spectral properties of many noise types found in nature signify fractal properties to a good approximation [53], [69]. This includes transmission noise over a range of radio and microwave spectra, for example, and Internet traffic noise [33]. With regard to Internet traffic noise, the time series data representing packet size and inter-arrival times shows well defined random fractal properties with a fractal dimension that varies over a 24 hour cycle. This can be used to submit emails by fracturing files into byte sizes that characterise packet size and submitting each fractured file at time intervals that characterise the inter-arrival times at the point of submission [98], [99]. In both cases, the principal 'characteristic' is the

fractal dimension computed from live Internet data.

E. Review of Encryption and Steganographic Models

The basic models considered in this paper relate to the seperate processes of encryption and steganography. For the symmetric encryption method considered, the encryption process is undertaken using the conventional result (given one-dimensional bit streams)

$$ciphertext = cipher \oplus plaintext.$$

For the steganographic and image watermarking methods considered, the model is (given two-dimensional floating point arrays)

$$stegotext = cipher \otimes plaintext + covertext$$

for electronic-to-electronic steganography and

$$ciphertext = cipher \otimes plaintext$$

for electronic-to-hardcopy steganography. In each case, the *cipher* can be generated through application of multi-algorithmic IFS as used in the product Crypstic$^{\rm TM}$, for example, reported in Section XII.

APPENDIX I
CENTRAL LIMIT THEOREM FOR A UNIFORM DISTRIBUTION

We study the effect of applying multiple convolutions of the uniform distribution

$$P(x) = \begin{cases} \frac{1}{X}, & \mid x \mid \leq X/2; \\ 0, & \text{otherwise} \end{cases}$$

and show that

$$\prod_{i=1}^{N}{\!\!\!\!\otimes}\ P_i(x) \equiv P_1(x) \otimes P_2(x) \otimes ... \otimes P_N(t)$$

$$\simeq \sqrt{\frac{6}{\pi N}} \exp(-6x^2/XN)$$

where $P_i(x) = P(x), \ \forall n$ and N is large. by considering the effect of multiple convolutions in Fourier space (through application of the convolution theorem) and then working with a series representation of the result.

The Fourier transform of $P(x)$ is given by

$$\widetilde{P}(k) = \int_{-\infty}^{\infty} P(x) \exp(-ikx) dx$$

$$= \int_{-X/2}^{X/2} \frac{1}{X} \exp(-ikx) dx = \text{sinc}(\text{kX}/2)$$

where $\text{sinc}(\text{x}) = \sin(\text{x})/\text{x}$ - the 'sinc' function. Thus,

$$P(x) \Longleftrightarrow \text{sinc}(\text{kX}/2)$$

where $\Longleftrightarrow$ denotes transformation into Fourier space, and from the convolution theorem in follows that

$$Q(x) = \prod_{i=1}^{N}{\!\!\!\!\otimes}\ P_i(x) \Longleftrightarrow \text{sinc}^{\text{N}}(\text{kX}/2).$$

Using the series expansion of the sin function for an arbitrary constant α,

$$\text{sinc}(\alpha\text{k}) = \frac{1}{\alpha\text{k}} \left(\alpha\text{k} - \frac{1}{3!}(\alpha\text{k})^3 + \frac{1}{5!}(\alpha\text{k})^5 - \frac{1}{7!}(\alpha\text{k})^7 + \ldots \right)$$

$$= 1 - \frac{1}{3!}(\alpha k)^2 + \frac{1}{5!}(\alpha k)^4 - \frac{1}{7!}(\alpha k)^6 + \ldots$$

The N^{th} power of $\text{sinc}(\alpha\text{k})$ can be written in terms of a binomial expansion giving

$$\text{sinc}^{\text{N}}(\alpha\text{k}) = \left(1 - \frac{1}{3!}(\alpha\text{k})^2 + \frac{1}{5!}(\alpha\text{k})^4 - \frac{1}{7!}(\alpha\text{k})^6 + \ldots \right)^{\text{N}}$$

$$= 1 - N\left(\frac{1}{3!}(\alpha k)^2 - \frac{1}{5!}(\alpha k)^4 + \frac{1}{7!}(\alpha k)^6 - \ldots \right)$$

$$+ \frac{N(N-1)}{2!} \left(\frac{1}{3!}(\alpha k)^2 - \frac{1}{5!}(\alpha k)^4 + \frac{1}{7!}(\alpha k)^6 - \ldots \right)^2$$

$$- \frac{N(N-1)(N-2)}{3!} \left(\frac{(\alpha k)^2}{3!} - \frac{(\alpha k)^4}{5!} + \frac{(\alpha k)^6}{7!} - \ldots \right)^3$$

$$+ \ldots$$

$$= 1 - N\frac{\alpha^2 k^2}{3!} + N\frac{\alpha^4 k^4}{5!} - k\frac{\alpha^6 k^6}{7!}$$

$$- \ldots + \frac{N(N-1)}{2!} \left(\frac{\alpha^4 k^4}{(3!)^2} - 2\frac{\alpha^6 k^6}{3!5!} + \ldots \right)$$

$$- \frac{N(N-1)(N-2)}{3!} \left(\frac{\alpha^6 k^6}{(3!)^3} + \ldots \right) + ...$$

$$= 1 - \frac{N}{3!}\alpha^2 k^2 + \left(\frac{N}{5!}\alpha^4 + \frac{N(N-1)}{2!(3!)^2}\alpha^4 \right) k^4$$

$$- \left(\frac{N}{7!}\alpha^6 + \frac{N(N-1)}{3!5!}\alpha^6 + \frac{N(N-1)(N-2)}{3!(3!)^3}\alpha^6 \right) k^6 + \ldots$$

Now the series representation of the exponential (for an arbitrary positive constant c) is

$$\exp(-ck^2) = 1 - ck^2 + \frac{1}{2!}c^2k^4 - \frac{1}{3!}c^3k^6 + \ldots$$

Equating terms involving k^2, k^4 and k^6 it is clear that (evaluating the factorials),

$$c = \frac{1}{6}N\alpha^2,$$

$$\frac{1}{2}c^2 = \left(\frac{1}{120}N + \frac{1}{72}N(N-1) \right) \alpha^4$$

or

$$c^2 = \left(\frac{1}{36}N^2 - \frac{1}{90}N \right) \alpha^4,$$

and

$$\frac{1}{6}c^3 = \left(\frac{N}{5040} + \frac{N(N-1)}{720} + \frac{N(N-1)(N-2)}{1296} \right) \alpha^6$$

or

$$c^3 = \left(\frac{1}{216}N^3 - \frac{1}{1080}N^2 + \frac{1}{2835}N\right)\alpha^6.$$

Thus, by deduction, we can conclude that

$$c^n = \left(\frac{1}{6}N\right)^n \alpha^{2n} + O(N^{n-1}\alpha^{2n}).$$

Now, for large N, the first term in the equation above dominates to give the following approximation for the constant c,

$$c \simeq \frac{1}{6}N\alpha^2.$$

We have therefore shown that the N^{th} power of the $\mathrm{sinc}(\alpha \mathrm{k})$ function approximates to a Gaussian function (for large N), i.e.

$$\mathrm{sinc^N}(\alpha \mathrm{k}) \simeq \exp\left(-\frac{1}{6}\mathrm{N}\alpha^2\mathrm{k}^2\right).$$

Thus, if $\alpha = \frac{X}{2}$, then

$$Q(x) \Longleftrightarrow \exp\left(-\frac{X}{24}Nk^2\right)$$

approximately. The final part of the proof is therefore to Fourier invert the function $\exp(-XNk^2/24)$, i.e. to compute the integral

$$I = \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp\left(-\frac{1}{24}XNk^2\right)\exp(ikx)dk$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp\left(-\left[\left(\sqrt{\frac{XN}{24}}k - \sqrt{\frac{24}{XN}}\frac{ix}{2}\right)^2 + \frac{6x^2}{XN}\right]\right)dk$$

$$= \frac{1}{\pi}\sqrt{\frac{6}{XN}}\mathrm{e}^{-\frac{6x^2}{XN}} \int_{-\infty+ix\sqrt{\frac{6}{XN}}}^{\infty+ix\sqrt{\frac{6}{XN}}} \mathrm{e}^{-y^2}dy$$

after making the substitution

$$y = \sqrt{\frac{XN}{6}}\frac{k}{2} - ix\sqrt{\frac{6}{XN}}.$$

By Cauchy's theorem

$$I = \frac{1}{\pi}\sqrt{\frac{6}{XN}}\mathrm{e}^{-\frac{6x^2}{XN}}\int_{-\infty}^{\infty}\mathrm{e}^{-z^2}dz = \sqrt{\frac{6}{\pi XN}}\mathrm{e}^{-\frac{6x^2}{XN}}$$

where we have use the result

$$\int_{-\infty}^{\infty}\exp(-y^2)dy = \sqrt{\pi}.$$

Thus, we can write

$$Q(x) = \prod_{i=1}^{N}{}_{\otimes}\ P_i(x) \simeq \sqrt{\frac{6}{\pi XN}}\exp[-6x^2/(XN)]$$

for large N.

APPENDIX II
THE LYAPUNOV DIMENSION

A principal and distinctive characteristic of a chaotic system is bifurcation and a common measure of this characteristic is the Lyapunov exponent. For iterative processes in general, where stable convergent behaviour is expected, an output that is characterised by exponential growth can be taken to be due to unacceptable numerical instability. However, with IFS that exhibit intrinsic instability leading to bifurcation and chaos where the output does not converge to a specific value, the Lyapunov exponent is used to quantify the characteristics of the output. This exponent or 'Dimension' provides a measure of 'chaoticity' and thus, how long before a 'forecast' becomes redundant.

Consider an iterative system defined by some (typically non-linear) function f of the form

$$x_{n+1} = f(x_n) = x + \epsilon_n$$

where ϵ_n is a perturbation to the value of x at an iterate n which is independent of the value of x_0. If the system converges to f as $n \to \infty$ then $\epsilon_n \to 0$ as $n \to \infty$ and the system is stable. If this is not the case, then the system may be divergent or chaotic. Suppose we model ϵ_n in terms of an exponential growth ($\lambda > 0$) or decay ($\lambda < 0$) so that

$$\epsilon_{n+1} = c\exp(n\lambda)$$

where c is an arbitrary constant. Then $\epsilon_1 = c$, $\epsilon_2 = \epsilon_1\exp(\lambda)$, $\epsilon_3 = \epsilon_1\exp(2\lambda) = \epsilon_2\exp(\lambda)$ and thus, in general, we can write

$$\epsilon_{n+1} = \epsilon_n\exp(\lambda).$$

Noting that

$$\ln\left(\frac{\epsilon_{n+1}}{\epsilon_n}\right) = \lambda$$

we can write

$$\sum_{n=1}^{N}\ln\left(\frac{\epsilon_{n+1}}{\epsilon_n}\right) = N\lambda.$$

Thus, we can define λ as

$$\lambda = \lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\ln\left(\frac{\epsilon_{n+1}}{\epsilon_n}\right).$$

The constant λ is known as the Lyapunov exponent. Since we can write

$$\lambda = \lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}(\ln\epsilon_{n+1} - \ln\epsilon_n)$$

and noting that (using forward differencing)

$$\frac{d}{dx}\ln\epsilon_n \simeq \frac{\ln\epsilon_{n+1} - \ln\epsilon_n}{\delta x} = \ln\epsilon_{n+1} - \ln\epsilon_n, \quad \delta x = 1$$

we see that λ is, in effect, given by the mean value of the derivatives of the natural logarithm of ϵ_n, i.e.

$$\lambda = \lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\left(\frac{d}{dx}\ln\epsilon_n\right)$$

Note that, if the value of λ is negative, then the iteration is stable and will approach f since we can expect that as $N \rightarrow \infty$, $\epsilon_{n+1}/\epsilon_n < 1$ and, thus, $\ln(\epsilon_{n+1}/\epsilon_n) < 0$. If λ is positive, then the iteration will not converge to x but will diverge or, depending on the characteristics of the function f, may exhibit bifurcation leading to chaotic behaviour. Since $f(x_n) = x + \epsilon_n$, whose behaviour is dependent on the initial condition x_0, we can consider the following definition of the Lyapunov exponent as a function of the initial condition:

$$\lambda(x_0) = \lim_{N\rightarrow\infty} \frac{1}{N} \sum_{n=1}^{N} \ln |f'(x_n)| = \lim_{N\rightarrow\infty} \frac{1}{N} \log \prod_{n=1}^{N} |f'(x_n)|$$

where f' denotes the derivative of f. In this form, we see that for each n, $f'(x_n)$ expresses how much the function f changes with respect to its argument at the point x_n. The derivative expresses the magnitude of change in transition from x_k to x_{k+1}. The limit is the average of the derivative logarithms over n iterations and provides a measure of how fast the IFS changes as a function of n. A positive Lyapunov exponent is therefore an indication of chaotic behavior and its magnitude, a measure of the extent of 'chaoticity'. For example, for the Logistic map

$$f(x) = 4rx(1-x), \quad x \in X = (0,1), \quad x_0 \in X, \quad r \in (0,1)$$

the Lyapunov exponent is given by

$$\lambda(x_0) = \frac{1}{N} \sum_{n=1}^{N} \log \mid r(1-2x_n) \mid$$

and for $r = 0.9$ and $N = 4000$,

$$\lambda(0.5) \approx 0.7095$$

In the application of chaos to cryptography, the Lyapunov exponent can be used to evaluate the sensitivity to initial conditions. For example, larger values of λ indicate that we need less encryption rounds in order to generate a cipher. Since the Lyapunov exponent is a characterization of the 'chaoticity' of the signal $f(x_n)$, if we compute λ_N using N elements of the signal f_n [i.e. N elements of the output from the ISF $x_{n+1} = f(x_n)$] and then compute λ_M using M elements of the same signal, we can define the Lyapunov dimension as

$$D_L = \begin{cases} 1 - \frac{\lambda_N}{\lambda_M}, & \lambda_M > \lambda_N; \\ 1 - \frac{\lambda_M}{\lambda_N}, & \lambda_M < \lambda_N. \end{cases}$$

ACKNOWLEDGMENT

Some of the results and ideas reported in this paper are based on the research and PhD Theses of the following former research students of the author: Dr Nasser Al-Ismaili, Dr Dmitri Dubovitski, Dr Khaled Mahmoud, Dr Rashiq Marie, Dr Sergei Mikhailov and Dr Nicholai Ptitsyn. The author is grateful to the following for their contributions to Crypstic$^{\mathrm{TM}}$ Limited in terms of its investment portfolio: Mr Dean Attew, Major General John Holmes, Mr Bruce Murray and Mr William Kidd.

REFERENCES

[1] S. Singh, *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots to Quantum Cryptography*, Doubleday, 1999.
[2] B. Schneier, *Beyond Fear: Digital Security in a Networked World*, Wiley, 2000.
[3] B. Schneier, *Thinking Sensibly about Security in an Uncertain World*, Copernicus Books, 2003.
[4] N. Ferguson and B. Schneier B, *Practical Cryptography*, Wiley, 2003.
[5] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 2001.
[6] B. Schneier, *Applied Cryptography*, Second Edition Wiley, 1996.
[7] J. Buchmann, *Introduction to Cryptography*, Springer 2001.
[8] O. Goldreich, *Foundations of Cryptography*, Cambridge University Press, 2001.
[9] J. Hershey, *Cryptography Demystified*, McGraw-Hill, 2003.
[10] H. F. Gaines, *Cryptanalysis*, Dover, 1939.
[11] N. V. Ptitsyn, *Deterministic Chaos if Digital Cryptography*, PhD Thesis, De Montfort University, 2003.
[12] http://vl.fmnet.info/safety/
[13] http://www.amazon.com/Network-Security-process-not-product
[14] http://en.wikipedia.org/wiki/Enigma_Machine
[15] S. Katzenbeisser and F. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, 2000.
[16] N. F. Johnson, Z. Duric and S. Jajodia, *Information Hiding: Steganography and Watermarking -Attacks and Countermeasures*, Kluwer Academicf Publishers, 2001.
[17] G. Kipper, *Investigators Guide to Steganography*, CRC Press, 2004.
[18] Hacker's Black Book, http://www.hackersbook.com
[19] A. N. Shulsky and G. J. Schmitt, *Silent Warfare: Understanding the World of Intelligence*, Brassey, 2002.
[20] R. Hough, *The Great War at Sea*, Oxford University Press, 1983
[21] P. G. Halpern, *A Naval History of World War One*, Routledge, 1994.
[22] R. A. Ratcliff, *Delusions of Intelligence*, Cambridge University Press, 2006.
[23] R. A. Woytak, *On the Boarder of War and Peace: Polish Intelligence and Diplomacy and the Origins of the Ultra-Secret*, Columbia University Press, 1979.
[24] W. Kozaczuk, *Enigma: How the German Machine Cipher was Broken, and how it was Read by the Allies in World War Two*, University Publications of America, 1984.
[25] B. Booss-Bavnbek and J. Hoyrup, *Mathematics at War*, Birkhäuser, 2003.
[26] B. J. Copelend, *Colossus: The Secrets of Bletchley Parks Code Breaking Computers*, Oxford University Press, 2006.
[27] A. Stripp and F. H. Hinsley, *Codebreakers: The Inside Story of Bletchley Park*, Oxford University Press, 2001
[28] http://www.gchq.gov.uk/
[29] W. R. Harwood, *The Disinformation Cycle: Hoaxes, Delusions, Security Beliefs, and Compulsory Mediocrity*, Xlibris Corporation, 2002.
[30] R. Miniter, *Disinformation*, Regnery Publishing, 2005.
[31] T. Newark and J. F. Borsarello, *Book of Camouflage* Brassey's, 2002.
[32] J. M. Blackledge, B. Foxon and M. Mikhailov, *Fractal Modulation Techniques for Digital Communications Systems*, Proceedings of IEEE Conference on Military Communications, October 1998, Boston, USA.
[33] J. M. Blackledge, S. Mikhailov and M. J. Turner, *Fractal Modulation and other Applications from a Theory on the Statistics of Dimension*, Fractal in Multimedia (Eds M F Barnsely, D Saupe and E R Vrscay), The IMA Volumes in Mathematics and its Applications, Springer, 2002, 175-195.
[34] H. Gerrad and P. D. Antill, *Crete 1941: Germany's Lightning Airborne Assault*, Osprey Publishing, 2005.
[35] http://eprint.iacr.org/1996/002.
[36] J. Buchmann, *Introduction to Cryptography*, Springer, 2001.
[37] H. Delfs and H. Knebl, *Introduction to Cryptography: Principles and Applications*, Springer, 2002.
[38] V. V. Ashchenko, V. V. Jascenko and S. K. Lando, *Cryptography: An Introduction*, American Mathematical Society, 2002.
[39] A. Salomaa, *Public Key Cryptography*, Springer, 1996.
[40] Articsoft Technologies, *Introduction to Encryption*, 2005; http://www.articsoft.com/wp_explaining_encryption.htm.
[41] C. Ellison and B. Shneier, *Ten Risks of PKI: What Your Not Being Told About Public Key Infrastructure*, Computer Security Journal XVI(1), 2000; http://www.schneier.compaper-pki.pdf.
[42] P. Garrett, *Making, Braking Codes*, Prentice Hall, 2001.
[43] P. Reynolds, *Breaking Codes: An Impossible Task?*, 2004; http://news.bbc.co.uk/1/hi/technology/3804895.stm.

[44] N. Al-Ismaily, *Dynamic Block Encryption with Self-Authenticating Key Exchange*, PhD Thesis, Loughborough University, 2006.
[45] A. G. Webster, *Partial Differential Equations of Mathematical Physics*, Stechert, 1933.
[46] P.M. Morse and H. Feshbach, *Methods of Theoretical Physics*, McGraw-Hill, 1953.
[47] E. Butkov, *Mathematical Physics*, Addison-Wesley, 1973.
[48] G.A. Evans, J. M. Blackledge and P. Yardley, *Analytical Solutions to Partial Differential Equations*, Springer, 1999.
[49] G. F. Roach, *Green's Functions (Introductory Theory with Applications)*, Van Nostrand Reihold, 1970.
[50] I. Stakgold, *Green's Functions and Boundary Value Problems*, Wiley, 1979.
[51] P. A. M. Dirac, *The Principles of Quantum Mechanics*, Oxford University Press, 1947.
[52] R. F. Hoskins, *The Delta Function*, Horwood Publishing, 1999.
[53] J. M. Blackledge, *Digital Image Processing*, Horwood Publishing, 2005.
[54] A. Papoulis, *The Fourier Integral and its Applications*, McGraw-Hill, 1962.
[55] R. N. Bracewell, *The Fourier Transform and its Applications*, McGraw-Hill, 1978.
[56] G. P. Wadsworth and J. G. Bryan, *Introduction to Probability and Random Variables*, McGraw-Hill, 1960.
[57] B. L. van der Waerden, *Mathematical Statistics*, Springer-Verlag, 1969.
[58] R. G. Laha and E. Lukacs, *Applications of Characteristic Functions*, Griffin, 1964.
[59] E. J. Watson, *Laplace Transforms and Applications*, Van Nostrand Reinhold, 1981.
[60] D. Wackerly, R. L. Scheaffer and W. Mendenhall, *Mathematical Statistics with Applications (6th Edition)*, Duxbury, May 2001.
[61] S. S. Wilks, *Mathematical Statistics*, Wiley, 1962.
[62] M. Born and E. Wolf, *Principles of Optics (6th Edition)*, Pergamon Press, Oxford, 1980.
[63] E. G. Steward, *Fourier Optics: An Introduction*, Horwood Scientific Publishing, 1987.
[64] M. V. Klein and T. E. Furtak, *Optics*, Wiley, 1986.
[65] E. Hecht, *Optics*, Addison-Wesley, 1987.
[66] I. J. Cox, M. L. Miller and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann, 2002.
[67] B. B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman, 1983.
[68] M. F. Barnsley, R. L. Dalvaney, B. B. Mandelbrot, H. O. Peitgen, D. Saupe and R. F. Mandelbrot, *The Science of Fractal Images*, Springer, 1988.
[69] M. J. Turner, J. M. Blackledge and P. R. Andrews, *Fractal Geometry in Digital Imaging*, Academic Press, 1997.
[70] C. E. Shannon, *A Mathematical Theory of Communication*, Bell System Technical Journal, **27**, 379-423 (July), 623-656 (October), 1948.
[71] J. Sethna, *Statistical Mechanics : Entropy, Order Parameters and Complexity*, Oxford University Press, 2006.
[72] B. B. Buck and V. A. Macaulay (Eds.), *Maximum Entropy in Action*, Clarendon Press, 1992.
[73] http://www.freedownloadscenter.com/Best/des3-source.html
[74] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[75] E. Beham, *Cryptanalysis of the Chaotic-map Cryptosystem*, Suggested at EUROCRYPT'91, Technical paper, 1991, http://citeseer.nj.nec.com/175190.html
[76] M. E. Bianco and D. Reed, *An Encryption System Based on Chaos Theory*, US Patent No. 5048086, 1991.
[77] L. J. Kocarev, K. S. Halle, K. Eckert and L. O. Chua, *Experimental Demonstration of Secure Communications via Chaotic Synchronization*, IJBC, **2**(3), 709-713, 1992.
[78] T. Caroll and L. M. Pecora, 1990, *Synchronization in Chaotic Systems*, Phys. Rev. Letters, **64**(8), 821-824, 1990.
[79] T. Caroll and L. M. Pecora, *Driving Systems with Chaotic Signals*, Phys. Rev. **A44**(4), 2374-2383, 1991.
[80] T. Caroll and L. M. Pecora, *A Circuit for Studying the Synchronization of Chaotic Systems*, Journal of Bifurcation and Chaos, **2**(3), 659-667, 1992.
[81] J. M. Carroll, J. Verhagen and P. T. Wong, *Chaos in Cryptography: The Escape From the Strange Attractor*, Cryptologia, **16**(1), 52-72, 1992.
[82] M. S. Baptista, *Cryptography with Chaos*, Physics Letters A, **240**(1-2), 50-54, 1998.
[83] E. Alvarez, A. Fernandez, P. Garcia, J. Jimenez and A. Marcano, *New Approach to Chaotic Encryption*, Physics Letters A, **263**(4-6), 373-375, 1999.
[84] L. Cappelletti, *An FPGA Implementation of a Chaotic Encryption Algorithm*, Bachelor Thesis. Università Degli Studi di Padova, 2000 http://www.lcappelletti.f2s.com/Didattica/thesis.pdf
[85] L. Kocarev, *Chaos-based Cryptography: a Brief Overview*, Journal of Circuits and Systems, **1**(3), 6-21, 2001.
[86] Y. H. Chu and S. Chang, *Dynamic cryptography based on synchronized chaotic systems*, Electronic Letters, **35**(12), 1999.
[87] Y. H. Chu and S. Chang, *Dynamic data encryption system based on synchronized chaotic systems*, Electronic Letters, **35**(4), 1999.
[88] F. Dachselt, K. Kelber and W. Schwarz, *Chaotic Coding and Cryptanalysis*, 1997, http://citeseer.nj.nec.com/355232.html
[89] J. Fridrich, *Secure Image Ciphering Based on Chaos*, Final Technical Report. USAF, Rome Laboratory, New York, 1997.
[90] J. B. Gallagher and J. Goldstein, *Sensitive dependence cryptography*, Technical Report, 1996, http://www.navigo.com/sdc/
[91] Gao, *Gao's Chaos Cryptosystem Overview*, Technical Report, 1996, http://www.iisi.co.jp/ppt/enggcc/
[92] N. V. Ptitsyn, J. M. Blackledge and V. M. Chernenky, *Deterministic Chaos in Digital Cryptography*, Proceedings of the First IMA Conference on Fractal Geometry: Mathematical Methods, Algorithms and Applications (Eds. J M Blackledge, A K Evans and M Turner), Horwood Publishing Series in Mathematics and Applications, 189-222, 2002.
[93] R. Matthews, *On the derivation of a chaotic encryption algorithm*, Cryptologia, (13): 2942, 1989.
[94] http://www.x-ways.net/winhex/
[95] http://www.wellresearchedreviews.com/computer-monitoring/
[96] http://www.ponemon.org/
[97] http://www.nist.gov/
[98] R. Marie, *Fractal-Based Models for Internet Traffic and their Application to Secure Data Transmission*, PhD Thesis, Loughborough University, 2007.
[99] R. Marie, J. M. Blackledge and H. Bez, *Characterisation of Internet Traffic using a Fractal Model*, Proc. 4th IASTED Int. Conf. on Signal Processing, Pattern Recognition and Applications, Innsbruck, 2007, 487-501.

Jonathan Blackledge received a BSc in Physics from Imperial College, London University in 1980, a Diploma of Imperial College in Plasma Physics in 1981 and a PhD in Theoretical Physics from Kings College, London University in 1983. As a Research Fellow of Physics at Kings College (London University) from 1984 to 1988, he specialized in information systems engineering undertaking work primarily for the defence industry. This was followed by academic appointments at the Universities of Cranfield (Senior Lecturer in Applied Mathematics) and De Montfort (Professor in Applied Mathematics and Computing) where he established new post-graduate MSc/PhD programmes and research groups in computer aided engineering and informatics. In 1994, he co-founded Management and Personnel Services Limited where he is currently Executive Director. His work for Microsharp (Director of R & D, 1998-2002) included the development of manufacturing processes now being used for digital information display units. In 2002, he co-founded a group of companies specializing in information security and cryptology for the defence and intelligence communities, actively creating partnerships between industry and academia. He currently holds academic posts in the United Kingdom and South Africa, and in 2007 was awarded Fellowships of the City and Guilds London Institute and the Institute of Leadership and Management together with Freedom of the City of London for his role in the development of the Higher Level Qualification programmes in Engineering, ICT and Business Administration, most recently, for the nuclear industry, security and financial sectors respectively. Professor Blackledge has published over one hundred scientific and engineering research papers and technical reports for industry, six industrial software systems, fifteen patents, ten books and has been supervisor to sixty research (PhD) graduates. He lectures widely to a variety of audiences composed of mathematicians, computer scientists, engineers and technologists in areas that include cryptology, communications technology and the use of artificial intelligence in process engineering, financial analysis and risk management. His current research interests include computational geometry and computer graphics, image analysis, nonlinear dynamical systems modelling and computer network security, working in both an academic and commercial context.